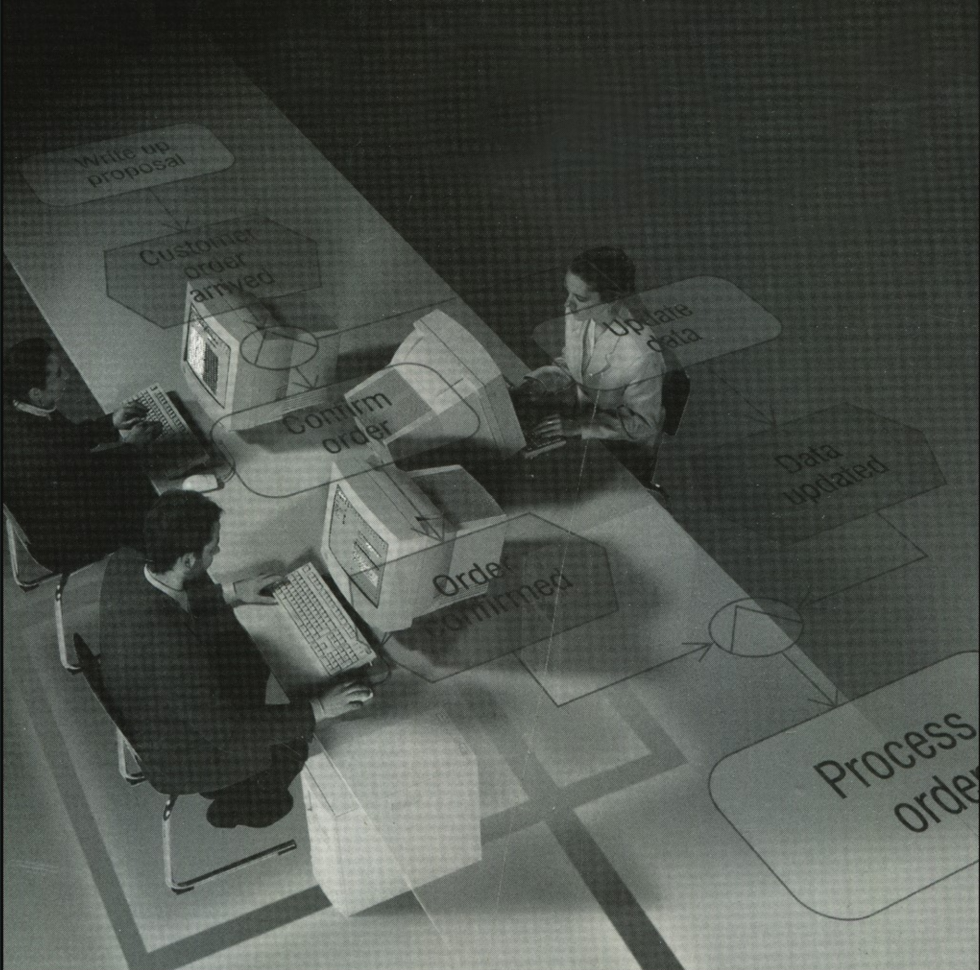
A.-W. Scheer

ARIS –
Business Process Frameworks

Second, Completely Revised
and Enlarged Edition



Springer

ARIS –
Business Process Frameworks

Second, Completely Revised and Enlarged Edition

Springer
Berlin
Heidelberg
New York
Barcelona
Budapest
Hong Kong
London
Milan
Paris
Singapore
Tokyo

August-Wilhelm Scheer

ARIS –

Business Process Frameworks

Second, Completely Revised and Enlarged Edition

With 94 Figures

Springer

Prof. Dr. Dr. h. c. August-Wilhelm Scheer
Universität des Saarlandes
Institut für Wirtschaftsinformatik
Postfach 15 11 50
D-66041 Saarbrücken
Germany

E-Mail: scheer@iwi.uni-sb.de
URL: http://www.iwi.uni-sb.de/info/scheer/scheer.html

The first edition of this book has been published under the title "Architecture of Integrated Information Systems". The second edition will be published in two volumes:
"ARIS - Business Process Frameworks" and
"ARIS - Business Process Modeling".

ISBN-13: 978-3-642-97740-4 e-ISBN-13: 978-3-642-97738-1
DOI: 10.1007/978-3-642-97738-1

Cataloging-in-Publication Data applied for
Die Deutsche Bibliothek - CIP-Einheitsaufnahme
Scheer, August-Wilhelm: ARIS - business process frameworks / August-Wilhelm Scheer. - 2., completely rev. and enl. ed. - Berlin; Heidelberg; New York; Barcelona; Budapest; Hong Kong; London; Milan; Paris; Singapore; Tokyo: Springer, 1998
ISBN-13: 978-3-642-97740-4

Hardcover-Design: Erich Kirchner, Heidelberg

SPIN 10678473 42/2202-5 4 3 2 1 0 - Printed on acid-free paper

Preface to the Second Edition

Since its first publication in 1992, the "Architecture of Integrated Information Systems" has been enjoying tremendous popularity. Documenting standard software with business models has proven to be a huge success. ARIS Toolset, developed by IDS Prof. Scheer GmbH and based on the ARIS concept, is now the worldwide leader in the market for business process engineering tools. Deployed in universities in the U.S., Europe, South Africa, Brazil and Asia-Pacific, ARIS Toolset is providing R&D and academic institutions engaged in enterprise organization and business information technology with a state-of-art business process engineering solution.

The furious development in information technology (IT) since the first edition of this book was published has led to so many new aspects and so much more information that we felt it necessary to completely revise it and actually split up the subject matter into two books, namely

ARIS - Business Process Frameworks and

ARIS - Business Process Modeling.

We see a different target audience for each book. Whereas the first book is aimed more at those interested in the business and design aspects of standard applications, the second book offers comprehensive insight into modeling and information technology.

About this Book

In "ARIS - Business Process Frameworks", we use the ARIS concept to describe business processes. The ARIS House of Business Engineering (HOBE) is a model for business process management.

The description of output flows is another new element in this edition. The HOBE concept now includes new software concepts such as workflow systems, componentware and frameworks. Instead of the entity relationship approach, we now employ the unified modeling language (UML) for describing meta models.

AD/CYCLE, which had originally seemed to be a promising approach, is no longer being sold by IBM, which is why we are no longer covering it.

The potential audience of this book includes IT (information technology) managers, consultants, instructors and students of business-related computer science, computer science and related disciplines. The author would be especially pleased if business administrations students were to regard this book as an enhancement to their discipline, in the sense of IT oriented business theory.

The illustrations in this book are available as slides on the WWW at http://www.iwi.uni-sb.de/lehre/aris-i/ and may be used, provided the copyright is observed and the source is mentioned.

I would like to express my gratitude to Mr. Christian C. Tiews of 'The Localizer' for the meticulous translation of the text into English. I would also like to thank Dipl.-Kff. Ursula Markus for coordinating and revising the translation of the German text into English, Dipl.-Kfm. Frank Habermann for his careful editing of the German manuscript, cand. rer. oec. Nathalie Anterist and cand. rer. inform. Jochen Kunze for the preparation of the English illustrations, and Prof. Thomas Gulledge from George Mason University, VA, for his careful revision of the English manuscript. Valuable technical input was provided by Dipl.-Wirtsch.-Ing. Markus Bold, Dr. Wolfgang Kraemer, Dipl.-Kfm. Markus Luzius, Dr. Markus Nüttgens and Dipl.-Ing. Arnold Traut.

Saarbrücken, Germany, July 1998

August-Wilhelm Scheer

Classification of the Contents

The books on business process engineering by this author adhere to a consistent principle, as depicted in Fig. I.

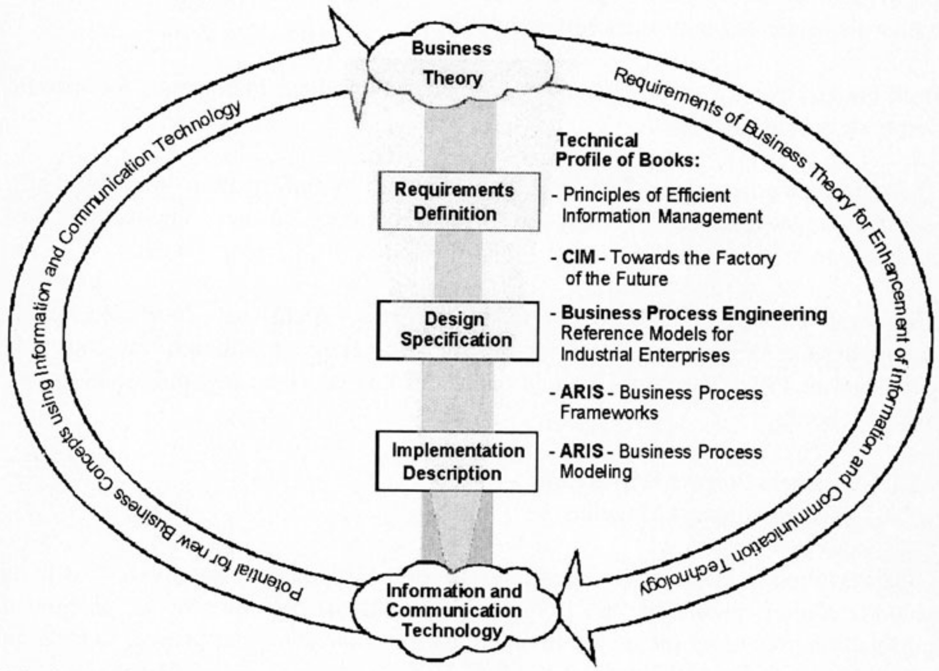


Fig. I: Technical profile of books by this author

Business-related computer science spans the gap between business theory -- and information and communication technology, with a bi-directional relationship between the two. Information and communication technology should be analyzed as to how new technical procedures can enable new IT oriented business application concepts. The "direction of influence" is illustrated by the arrow on the left hand side of Fig. I. In business-related computer science, it is not essential to know the full range of information technology, but only to apply the segment responsible for alterations in business application concepts. Business-related computer science is especially important in this area.

The arrow on the right hand side of Fig. I makes clear how the enhancement of information and communication technology is influenced by business requirements.

Both relationship directions are discussed in the book "Principles of Efficient Information Management", the second edition of which was published in 1991.

The key effects of information technology on business processes are discussed in "CIM (Computer Integrated Manufacturing) - Towards the Factory of the Future" which also appeared in its third edition in 1994.

Both books cover IT oriented frameworks and are excellent foundations for specific corporate system solutions.

These frameworks are implemented into IT tools by information systems. Thus, information systems really do act as bridges between business applications and information technology.

The "Architecture of Integrated Information Systems - ARIS" was developed for the comprehensive description of information systems. The first edition of the book was published in 1992. This is the second edition of this concept, now published in two different books,

ARIS - Business Process Frameworks -- and
ARIS - Business Process Modeling.

"Business Process Engineering - Reference Models for Industrial Enterprises", with its second edition published in 1994 offers industrial enterprises an integrated information system by the use of function, data, organization and process models, in accordance with the ARIS concept.

The business value of describing information systems decreases as technical implementation progresses. At the same time, stability of the concepts also diminishes because the enormous speed with which IT is being enhanced usually influences the technical implementation of information systems. In all of these books, the author takes this fact into account by the extent with which the respective issues are weighted. This is analogous to the weighting illustrated by the triangle in Fig. I.

All of the author's books are also available in German. "Business Process Engineering" is available in Chinese, "CIM" has been translated into Portuguese as well. Other translations are in progress.

Table of Contents

Abbreviations

ALE	Application Link Enabling
API	Application Programming Interface
ARIS	Architecture of Integrated Information Systems
BAPI	Business Application Programming Interface
BD	Business Data
BE	Business Engineer
BM	Business Management
BPO	Business Process Optimization
BPR	Business Process Reengineering
CAD	Computer Aided Design
CBO	Common Business Object
CIM	Computer Integrated Manufacturing
CIMOSA	Open System Architecture for Computer Integrated Manufacturing
CNC	Computerized Numerical Control
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CPI	Continuous Process Improvement
DCOM	Distributed COM
DIN	German Institute for Standards
EDI	Electronic Data Interchange
EIS	Executive Information System
EPC	Event Driven Process Chain
EQA	European Quality Award
ERM	Entity Relationship Model
ESHQ	Environment, Safety, Health and Quality
EU	European Union
GUI	Graphical User Interface
HOBE	ARIS - House of Business Engineering
HTML	Hypertext Markup Language
ICT	Information and Communication Technology
IDA	Interactive High Level Petri Nets
IDL	Interface Definition Language
IEM	Information Engineering Methodology
IFIP	International Federation for Information Processing
IMG	Implementation Management Guide
IS	Information Systems

ISA	Information System Architecture
ISDM	Information System Design Methodologies
ISM	Information System Methodology
ISO	International Organization for Standardization
IT	Information Technology
IWi	Institute for Information Systems, Saarbrücken, Germany
JSD	Jackson System Development
JVM	Java Virtual Machine
KBSt	Coordinating and Counseling Office of the German Government for Information Technology in Federal Administration
MMS	Merchandise Management System
MR	Microsoft Repository
NIAM	Nijssen Information Analysis Method
OAG	Open Application Group
OLAP	Online Analytical Processing
OMA	Object Management Architecture
OMG	Object Management Group
ORB	Object Request Broker
PSA	Problem Statement Analyzer
PSL	Problem Statement Language
QC	Quality Control
QM	Quality Management
RFC	Remote Function Call
SADT	Structured Analysis and Design Technique
SME	Small and Medium Sized Enterprises
SOM	Semantic Object Model
TQM	Total Quality Management
UML	Unified Modeling Language
VDA	Association of German Automobile Manufacturers
WAPI	Workflow Application Programming Interface
WfMC	Workflow Management Coalition
WMS	Workflow Management System

Table of Figures

A User Benefits of ARIS

ARIS stands for "Architecture of Integrated Information Systems". "Architecture" is a designation generally used in building construction. In information technology (IT), it describes the

⇒ type,
⇒ functional properties and the
⇒ interrelationship among

the individual building blocks of the information system.

The word "architecture" is quite common in IT concepts. Several authors have attempted to explain the shift in the usage of this designation from building construction to IT etymologically, such as Krcmar (*see Krcmar, Informationssystem-Architekturen 1990, p. 396)* and Strunz *(see Strunz, Informations- und Kommunikationssysteme 1990, p. 441*). However, this author feels that the shift in its usage is a result of colloquialism, rather than of etymology. In its IT-defined usage, also very common in U.S. publications, "architecture" is associated with other designations such as planning, tracking of rules, structuring or coordinating multiple business partners as well -- all of which are typical issues in information systems. As such, the designation "architecture" is also used to describe hardware and database systems (*see Lockemann/Dittrich, Architektur von Datenbanksystemen 1987, p. 87*).

In the first edition of this book, ARIS was used to develop a framework for describing (modeling) computer-aided information systems in their entirety -- from the requirements definition to the implementation description. We focused primarily on information systems and how they support business processes.

Today, due to an ever closer alignment of information technology to business processes, the significance of business process modeling has grown, with even classic business administration topics such as activity based costing, process (re)organization and quality management utilizing business process models in accordance with the ARIS concept. This results in business process modeling being increasingly regarded as an extension of business administration.

The terminology used in business administration unfortunately has certain disadvantages -- frequently being ambiguous, not sufficiently precise to finalize the issues at hand, and sometimes even leading to apparent contradictions. Therefore, oral descriptions are not especially suitable for specifying information systems. On the other hand, mathematical languages for describing decision and

planning issues in business administration are more accurate and easier to verify, yet they are not suitable for describing every issue.

This is why ARIS modeling methods also provide semi-conceptual methods of describing process-organizational issues. They coincide with business administration points of view, yet are also sufficiently precise and detailed -- providing an excellent starting point for further IS processing.

Thanks to new methods and tools for generating, customizing and configuring models, business process models are increasingly becoming a focal point and blueprint for configuring information systems. For this reason, the business-related description level in the ARIS concept has priority over implementation issues.

Semi-conceptual graphic methods, such as org charts or network diagrams, are quite common in business administration. However, the ARIS concept goes yet a step further, providing a reference guide for systematic and complete business process modeling.

Today, the description of business processes is increasingly being integrated with corporate documentation of enterprise-related knowledge or skills. "Knowledge management" and "organizational memory" are just two of the common terms in this context. Thus, the significance of ARIS as a framework for knowledge management continues to grow.

The ARIS concept primarily aids in capturing a wide range of descriptive aspects of business processes, allocating methods to them, determining any method overlap and identifying unoccupied description fields. ARIS provides advantages for solving business administration or organizational issues and for engineering computer-aided information systems as well.

A.I Benefits for Business Administration and Organizational Processes

Corporate mission statements entail the production and utilization of material output and services by combining production factors (*see Gutenberg, Die Produktion 1983, pp. 1-10*). Multiple entities and devices are responsible for fulfilling these tasks. In accordance with corporate objectives, their close collaboration must be ensured. The necessary rules for realizing the corporate objective are what we call an organization (*see Frese, Grundlagen der Organisation 1995, p. 1)*.

Structural or hierarchical organizations are characterized by time-independent (static) rules, such as by hierarchies or enterprise topologies. Here, relationships involving management, output, information or communication technology between departments, just to name a few, are entered. Org charts are some of the simple tools used to depict these relationships.

Process organizations, on the other hand, deal with time-dependent and logical (dynamic) behavior of the processes necessary to complete the corporate mission.

Hierarchical and process organizations are closely correlated.

For years, hierarchical organizations have been a key topic in business theory. However, due to buzzwords such as business process engineering *(see Gaitanides, Prozeßorganization 1983; Eversheim, Prozeßorientierte Unternehmensorganisation 1994; Nippa/Picot, Prozeßmanagement und Reengineering 1996)*, process organizations have moved into the spotlight in recent years.

Generally speaking, a business process is a continuous series of enterprise tasks, undertaken for the purpose of creating output. The starting point and final product of the business process is the output requested and utilized by corporate or external "customers".

Business processes often enable the value chain of the enterprise as well as focusing on the customer when the output is created. The purpose is to make the business process as significant as possible and to link multiple functions with it *(see Hammer/Champy, Business Reengineering 1995)*.

Business processes are always focused on business administration issues. Their goals can be strictly defined (such as reducing the lead time of the business process "product development" by 30%) and they are objects of cost accounting (activity based costing).

Many features in business process engineering or business process reengineering (BPR), respectively, have already been embraced by previous organizational concepts. For example, the Y-CIM model (*see Scheer, CIM 1994;* CIM = computer integrated manufacturing) is a concept describing the context between product development and the logistics process in an industrial enterprise, especially focused on the organization of business processes. As far back as 1984, the author described business processes by means of process chain diagrams, implementing these business processes as well *(see Scheer, Efficient Information Management 1991)*.

There are multiple reasons for creating business process models, such as:

- Optimizing organizational changes, a by-product of BPR,
- Storing corporate knowledge, in reference models, for example,
- Utilizing process documentation for ISO-9000 and other certifications,
- Calculating the cost of business processes,
- Leveraging process information to implement and customize standard software solutions or workflow systems.

Within these categories, other goals can be established for the modeling methods. In business process reengineering (BPR), we must therefore identify the components that need to be addressed.

These are some of the many issues that can be addressed by business process optimization:

- Changing the process structure by introducing simultaneous tasks, avoiding cycles, streamlining the structure,
- Changing organizational reporting structures and developing employee qualification by improving processing in its entirety,
- Reducing the amount of documentation, streamlining and accelerating document and data flow,
- Discussing possible outsourcing measures (shifting from internal to external output creation),
- Implementing new production and IT resources to improve processing functions.

In these examples, we are referring to numerous modeling aspects, such as process structures, hierarchical organizations, employee qualification, documents (data), external or internal output as well as production and IT resources. Obviously, a business process model for the purpose of optimization must be fairly complex. Moreover, it should address multiple aspects, for which numerous description methods are necessary. These various purposes determine the kind of modeling objects as well as the required granularity.

In addition to the actual purpose of modeling, the methods applied also determine the focus of the model. This is particularly true when a certain type of method has already been defined. Models reproduce excerpts of reality. They are created by abstracting the properties of real objects, whereas their essential structures and behavior remain intact (homomorphy). Not only the content-related purpose of the model, but also the permissible illustration methods determine to what extent non-essential characteristics may be abstracted. For example, if an object oriented modeling approach (e.g., the Petri net method) or a system-theoretic approach are selected, modeling only leads to objects applicable to the syntax or the semantics of these particular methods.

In order to avoid "lock-in" by certain methods, ARIS concepts are developed independently of any particular method, while supporting a generic business process definition. In particular, the ARIS concept excels in creating enterprise-wide business and production models.

A.II User Benefits for Developing Information Systems

Information systems can be designed as custom applications or purchased as off-the-shelf standard solutions. After the initial popularity of custom applications, integrated standard solutions are now the norm. With the advent of new types of software, such as componentware -- where software components for certain application cases are assembled to form entire applications -- , a blend between the two approaches has recently been making inroads. ARIS supports all of these: custom applications, standard software solutions and componentware assembly.

The development of custom applications is generally expensive and is often plagued by uncertainties, such as the duration of the development cycle or the difficulty of assessing costs. Thus, the tendency to shift software development from individual development to an organizational form of industrial manufacturing – in "software factories" -- is not surprising *(see Balzert, Entwicklung von Software-Systemen 1992, pp. 5)*.

In this context, multiple methods for supporting the software development process have been developed. They differ according to their focus on the various software development processes and according to their preferred approach regarding the issue at hand, such as data, event or function orientation, respectively. Miscellaneous works on software engineering give an overview of the numerous methods available. Here is a small sample of some standard works: Balzert *(Balzert, Lehrbuch der Software-Technik 1996)*, Sommerville *(Sommerville, Software Engineering 1987)* or the conference reports of the Working Group 8.1, published by the IFIP *(see z. B. Olle/Sol/Tully, Information Systems Design Methodologies 1983; Olle/Verrijn-Stuart/Bhabuta, Information Systems Life Cycle 1988; also see Preßmar/Eggers/Reinken, Interaktive Entwurfsmethode 1989; Barker, CASE* Method 1990; Hildebrand, Software Tools 1990)*.

Due to the wide range of methods differing only slightly from one another, this market is cluttered. In fact, the multitude of products and approaches has actually impeded the development of computer-aided tools based on these methods. We therefore provide a methodology (study of methods), covering various development methods.

The following are typical questions which leverage the framework capabilities of this methodology *(see Sol, Information Systems Design Methodologies 1983, p. 4; Olle et al., Information Systems Methodologies 1991, p. 2; Brodie/Ridjanovic/ Silva, Framework for Information Systems 1983, p. 232):*

1. Are there really so many totally different ways of designing a computer-aided information system?
2. If not, how similar are these methods? If so, why are there so many different ways?
3. Is there an optimal way of developing an information system?
4. Where does the development process start and where does it end?
5. What does the finished product of the design process look like?
6. How many steps are necessary to obtain a development result?
7. Should only one particular kind of information system be used or are several methods required, each for a different system? According to which criteria should the methods be selected?

The purpose of these questions is to classify and evaluate the various methods. After addressing these issues, there is, however, a second group of reasons for dealing with information system design methodologies (ISDM), resulting from the fact that, usually, several business partners are involved in complex development projects. Sometimes they use different development methods, or the results of their work might overlap. Only a framework integrating the individual methods, confirming agreement or pointing out any overlap, can lead to mutual understanding. Obviously, this framework can and should also coordinate the various methods. Unfortunately, many of today's popular methods for developing information systems seem more like fads, rather than empirical concepts based on sound theories.

The ARIS concept creates a guideline for developing, optimizing and implementing integrated application systems. At the same time, it shows business administration specialists how to view, analyze, document, and implement information systems.

Business administration software solutions comprise modules for accounting, purchasing, sales, production planning, etc. Financial information systems are characterized by an markedly high degree of complexity. Many corporate and external business partners are involved in the implementation of information systems. This becomes apparent in light of seamlessly integrated data processing, where data is shared by multiple applications. Some examples are comprehensive IS-oriented concepts implemented in enterprises, CIM in manufacturing companies, IS-supported merchandise management systems for retailers, or electronic banking in financial institutions.

Up to the mid '90s, the ratio between the effort of implementing financial packaged applications in organizations and their purchase price was frequently

more than 5:1. The reason for this high ratio is due to the fact that off-the-shelf systems are more or less easy to install, yet users must also determine which goals (strategies) they wish to reach with the system, how the functionality of the system can achieve this, and how to customize, configure and technically implement the package.

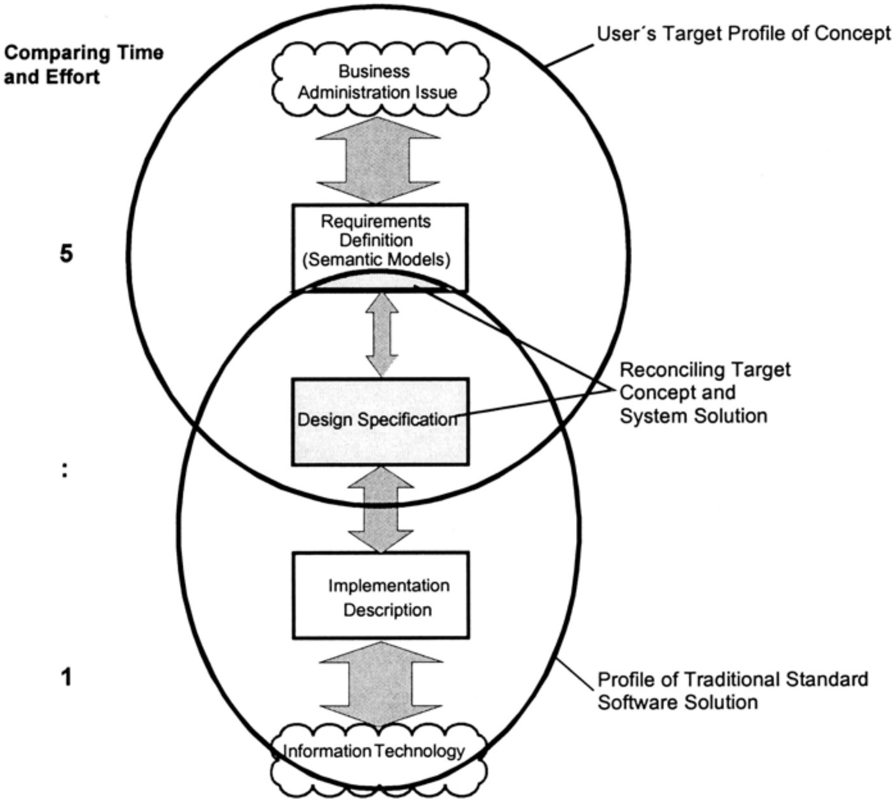


Fig.1a Comparing organizational effort and software effort within the life cycle

In the early life cycle model depicted in Fig.1a, software systems represented only the lower levels of the diagram. Business functionality in the system was depicted by "user interfaces, data tables, parameter settings, transaction names" or had to be derived accordingly. This is why users originally had to develop their own business requirements and reconcile the design specification with the standard software solution. This obviously required considerable IS skills and knowledge on how to execute business requirements, leading to users frequently being forced to rely on consultants.

With hardware and software costs rapidly decreasing, that ratio became even worse. Small and medium sized enterprises (SME) are not able to pay

consultancies millions of dollars for implementation. Hence, frameworks, methods and tools have become increasingly popular because they can help reduce the cost of software implementation and at the same time increase user acceptance of standard software solutions.

Several approaches are possible (see Fig.1b):

- Reduce the effort necessary for creating the target concept by leveraging "best practice case" knowledge available in reference models.
- Create a requirements definition by leveraging modeling techniques to detail the description.
- Document the requirements definition of the standard software by means of semantic modeling methods, making the business logic more understandable.
- Use semantic models to automate reconciliation of the requirements definition of the target concept with the standard software as much as possible, cutting down on the need for specific IS skills.
- Leverage semantic models as a starting point for maximum automation of system and configuration customizing.

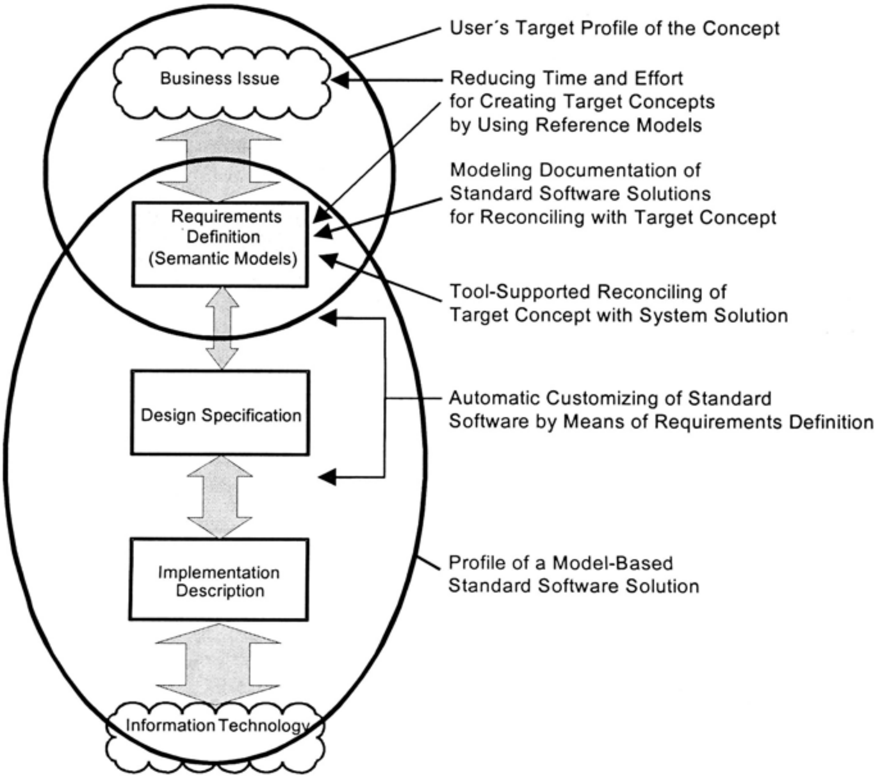


Fig.1b Various approaches for reducing the organizational effort

The ARIS concept was designed to include integrated methods and tools (ARIS Toolset) in order to leverage these approaches, speed up implementation and reduce the implementation effort for the standard software.

ARIS functionality

- offers a framework (architecture) to completely describe standard software solutions,
- integrates the most suitable methods for modeling information systems into the architecture, develops methods for describing business processes,
- provides reference models as tools for the administration of application know-how, for modeling and analyzing system requirements, and for ensuring user-friendly navigation within the models.
- leveraging standard software solutions, the ARIS house of business engineering (HOBE) provides an architecture for managing business processes. Using workflow systems, it is loosely linked with software "building blocks" (business objects). ARIS provides the framework for describing the assembly of the software components, resulting in business information systems which are ideal for configuring workflow systems, generating screens and determining parameters for applications.

B Basic Business Process Model in ARIS

In order to develop the ARIS concept, the respective object must first be studied closely, leading in turn to the creation of a simple business process model, primarily based on business expertise. This model is then enhanced by additional details, leading to the basic business process model in ARIS.

B.I The Initial Business Process Model

We explain the key issues in describing business processes with a simple example from customer order processing. First, let us outline the scenario:

Let us imagine a customer wants to order several items which need to be manufactured. Based on customer and item information, the feasibility of manufacturing this item is studied. Once the order has arrived, the necessary materials are obtained from a supplier. After arrival of the material and subsequent order planning, the items are manufactured according to a work schedule and shipped to the customer, along with the appropriate documentation.

This scenario is now discussed from various points of view.

In system theory, we can distinguish between system structures and system behavior. We will begin by describing the responsible entities and relationships involved in the business process, then, by means of function flow, we will describe the dynamic behavior. Output flows describe the results of executing the process, information flows illustrate the interchange of documents involved in the process.

Functions, output producers (organizational units), output and information objects are illustrated by various symbols. Flows are depicted by arrows.

B.I.1 Responsible Entities and their Relationships

Fig. 2a depicts the responsible entities (organizational units) involved in the business process, along with their output and communication relationships, illustrated as context or interaction diagrams. The sequence in which processes are carried out, is not apparent. Nevertheless, this provides an initial view of the business process structure. In complex processes, the myriad interchanges among the various business partners can become somewhat confusing. In addition to the

various interactions, it is also possible to enter the activities of the responsible entities. This has been done in only a few places.

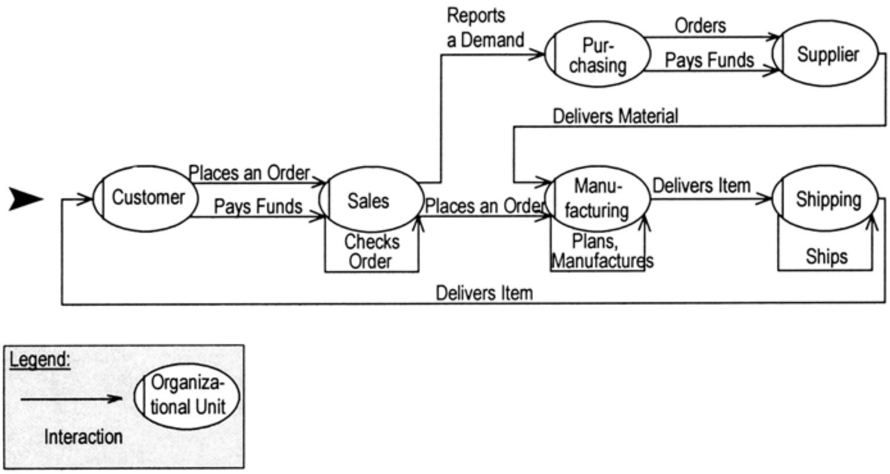


Fig. 2a Interaction diagram of the business process "order processing"

Interaction diagrams are quite common in business theory. This general illustration, depicted in Fig. 2b, of output and communication relationships among the customer, enterprise and supplier is quite typical.

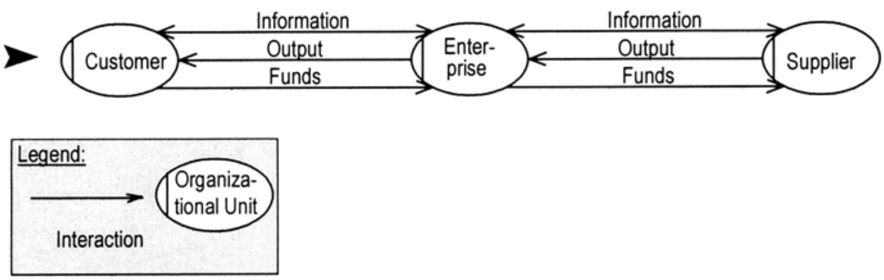


Fig. 2b General business interaction diagram in enterprises

B.I.2 Function Flow

Fig. 3 describes the same business process by depicting the activities (functions) to be executed, as well as their sequence. The main issue is not responsible

entities, as was the case with the static interaction diagram, but rather the dynamic sequence of activities. For illustration purposes, the operational units are also depicted in Fig. 3. Due to redundancies, their interrelationship with the interaction diagram is not as obvious.

Being function sequences for creating output, function flows characterize the business process. The output flows themselves will be displayed individually.

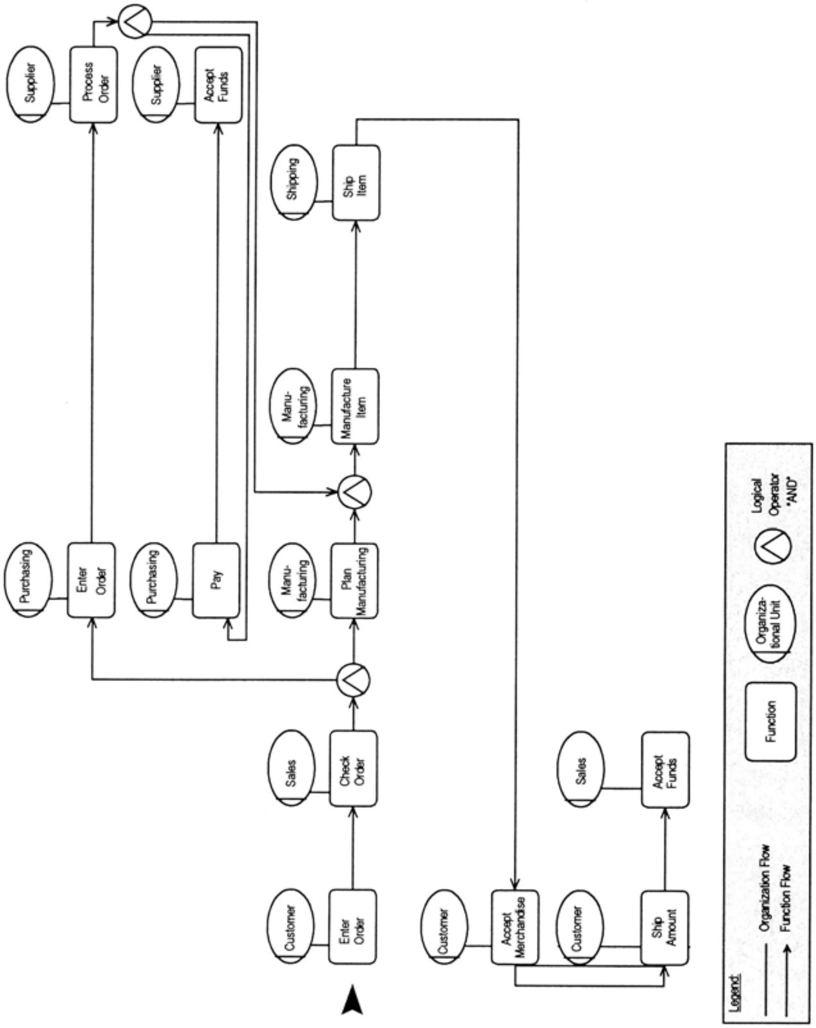


Fig. 3 Function flow of the business process „order processing"

B.I.3 Output Flow

The purpose of a business process is to create output in order to get rewarded by another output. In our example, the output of the enterprise is to execute the customer order and the reward is the receipt of funds, although within the enterprise, intermediate output as a result of executing the functions is also created.

The designation "output" is very heterogeneous. Business output is the result of a production process, in the most general sense of the word. Output can be physical (material output) or non-physical (services). Whereas material output is easily defined, for example by the delivery of material, manufactured parts or even the finished product, the term "services" is more difficult to define because it comprises heterogeneous services. These are some of the many kinds of services:

- Theatrical performances (theaters, concerts), where the output is the act of performing; consumption of this output is simultaneous with its production;
- Banks providing services in the form of loans or credits; in this case, the service consists of providing the necessary funds and is in itself the *result* of other banking services, such as credit checks, depositary services, etc.;
- Insurance services;
- Services in the public sector (issuing drivers licenses, IDs, etc.).

Some important characteristics of output are that they be required by a party other than the party providing them, i.e., there must be a demand for this output. Furthermore, they must be requested by the party using them and a price must be agreed upon. It makes no difference whether this customer-supplier relationship exists between external business partners or between internal organizational units. The respective price can be a market price or just involve inter-company invoicing. Furthermore, it is irrelevant whether the amount is actually asked for or paid. If the party purchasing the service recognizes the monetary value of the service, this is sufficient. Thus, inter-company services are sometimes free of charge, just as some external services in the public sector.

However, in order to improve the transparency of output, there is a tendency to define inter-company output and charge the resulting costs as well. This is also the case in the public sector. Functions creating output are depicted below the output symbols in Fig. 4. This output consists of information services, such as "checked order", "manufacturing plan", "order", "order documents" or "shipping order". On the other hand, items directly resulting from manufacturing are regarded as material output. The delivered items are the result of a "transportation" service.

The result of the process "manufacture item" in Fig. 4 is the material output, defined by the manufactured item. Likewise, quality checks are carried out and

documented during the manufacturing process. All data pertinent to the customer is captured in "order documents", themselves a service by means of the information they provide. After every inter-company function, output describing the deliverable is defined, in turn entering the next process as input. To avoid cluttering the diagram, the organizational units involved are not depicted. It is not possible to uniquely derive the function sequence from the illustration of the output flow.

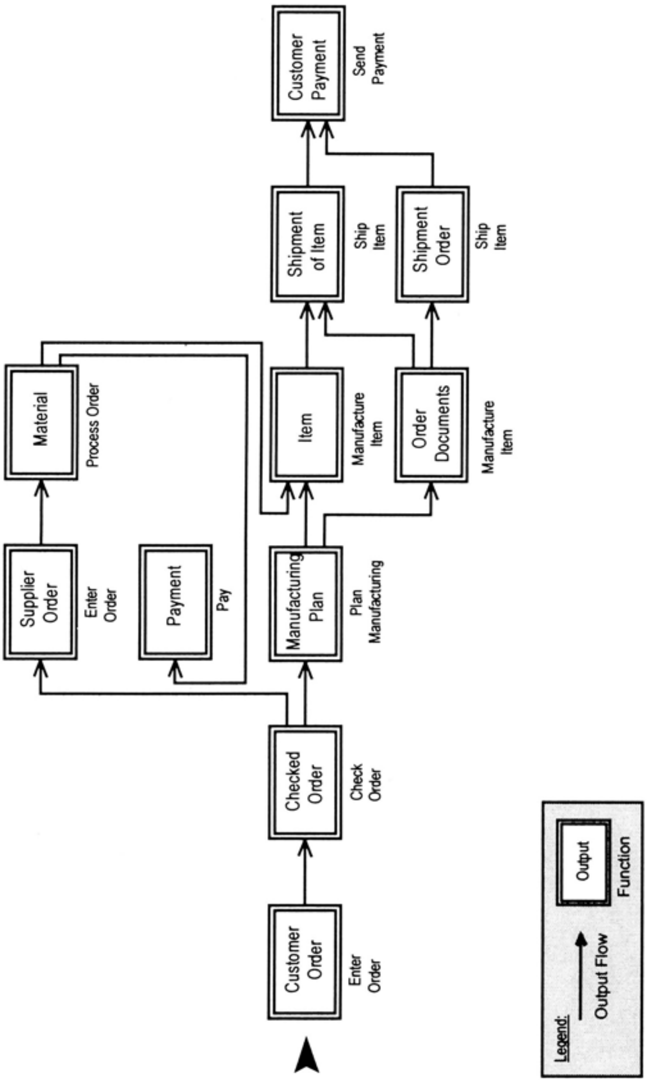


Fig. 4 Output flow of the business process „order processing“

B.I.4 Information Flow

In addition to information services, also other information, used as environment descriptions during the business processes, are process components.

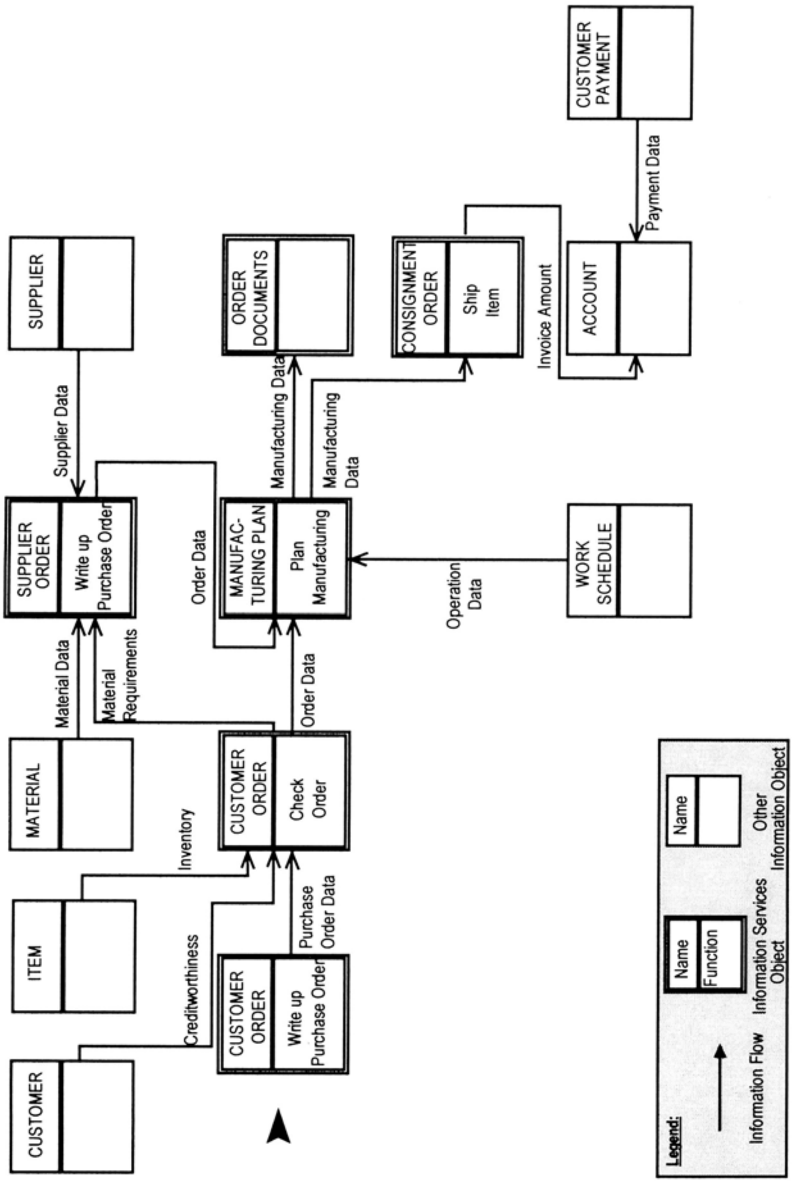


Fig. 5 Information flow of the business process „order processing"

Fig. 5 illustrates the information objects of the business process, along with the data interchanged among them. Objects listed as information services have double borders. Information objects describing the environment of the business process are shown as well, for example data regarding suppliers, items or work schedules. This data is necessary to create information services. For example, when checking orders, the customer's credit is checked and inventory is checked for availability.

Information objects are characterized by their names and other attributes can be assigned to them. However, in order to save space, the attributes have been omitted here. Functions of the process applied to the information objects have been allocated to the information services objects. Functions could also be allocated to other information objects. However, these would be sub-functions of the process functions discussed so far, which is why they are not illustrated in Fig. 5. For example, the detail function "carry out credit check" can be applied to the information object CUSTOMER, contained in the process function "check order".

Due to the fact that data flow is triggered by the functions that are linked to the information objects, it is more or less possible to read the function flow in Fig. 5. However, if multiple functions are applied to an information object or if multiple data flows are requested by a function, the function process cannot be uniquely deduced.

B.I.5 Consolidated Business Process Model

Note that none of the flows (organization, function, output and information flow, respectively) illustrated here are capable of completely modeling the entire business process. We must therefore combine all these perspectives. To this end, one of the views should be selected as a foundation and then be integrated into the others.

Because the function flow is closest to the definition of the business process, we will use it as a starting point (Fig. 6). When dealing with object oriented approaches at a later time, we will demonstrate how information flows can serve as starting points as well.

In order to differentiate relationships, we will illustrate the flows in various ways.

Although the arrows depicting function flow and output flow seem redundant, they do not always run parallel. After the function "process order", the function "manufacture item" is triggered by the supplier. Simultaneously, the supplier is paid by purchasing, although the former does not actually receive the physical output.

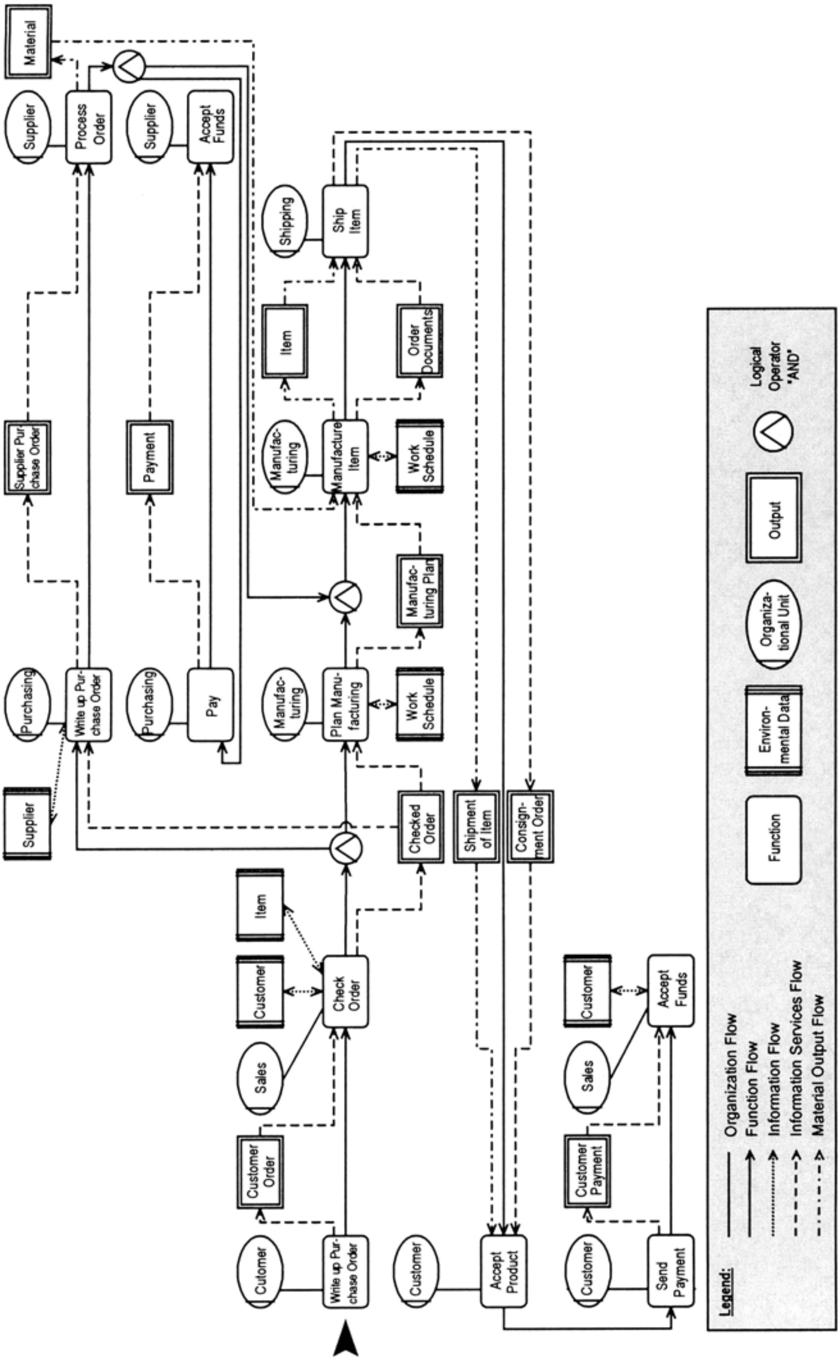


Fig. 6 Consolidated business process model „order processing“

Information objects already illustrated as services are not linked to the functions as information objects. If flows run parallel, such as output flows and function flows, the illustration can be streamlined by omitting one of them.

B.II The ARIS Business Process Model

The simple model depicted in Fig. 6 already agrees with the definition of business processes, although we will provide further details to make it more realistic. Subsequently, we will generalize the business process already discussed in the order processing example.

B.II.1 The Expanded Example Process

Fig. 7 depicts a more detailed version of the process "manufacture item" shown in Fig. 6. It illustrates the key enhancements of the semantic background of business processes used in this book. They are derived from business administration literature, business-related computer science and hands-on experience.

Function flows are enhanced by event and message controls. This makes it possible to better describe the process sequence. Events describe condition changes and, for example, characterize the beginning of the result of an event, in turn triggering the next function. In addition to simple events, there are also compounded events. For example, for the function "manufacture item", planning needs be concluded and the necessary parts need to be available. This is expressed by the logical "AND" operator between the events.

Control flows regulate how events are triggered in accordance with a sensible process logic. Sequential, parallel, alternative and merged methods, along with logical links, may be used. Control flows are executed by events and messages that they trigger, after which information regarding the beginning of the event is transferred to the next entity. In the illustrations, messages are depicted by letter symbols. They determine how functions react to events. Messages can also contain additional attributes besides information regarding the beginning of the event *(see Scheer, ARIS - Business Process Modeling 1998)*.

After the events "manufacturing plan completed" and "(supplier) order processed" have triggered the function "manufacture item" by means of messages, the event "item completed" is created. Then the process is concluded. This event activates successive events by means of messages.

Only events pertinent for the continuation of the business process are illustrated here. These events are known as relevant events.

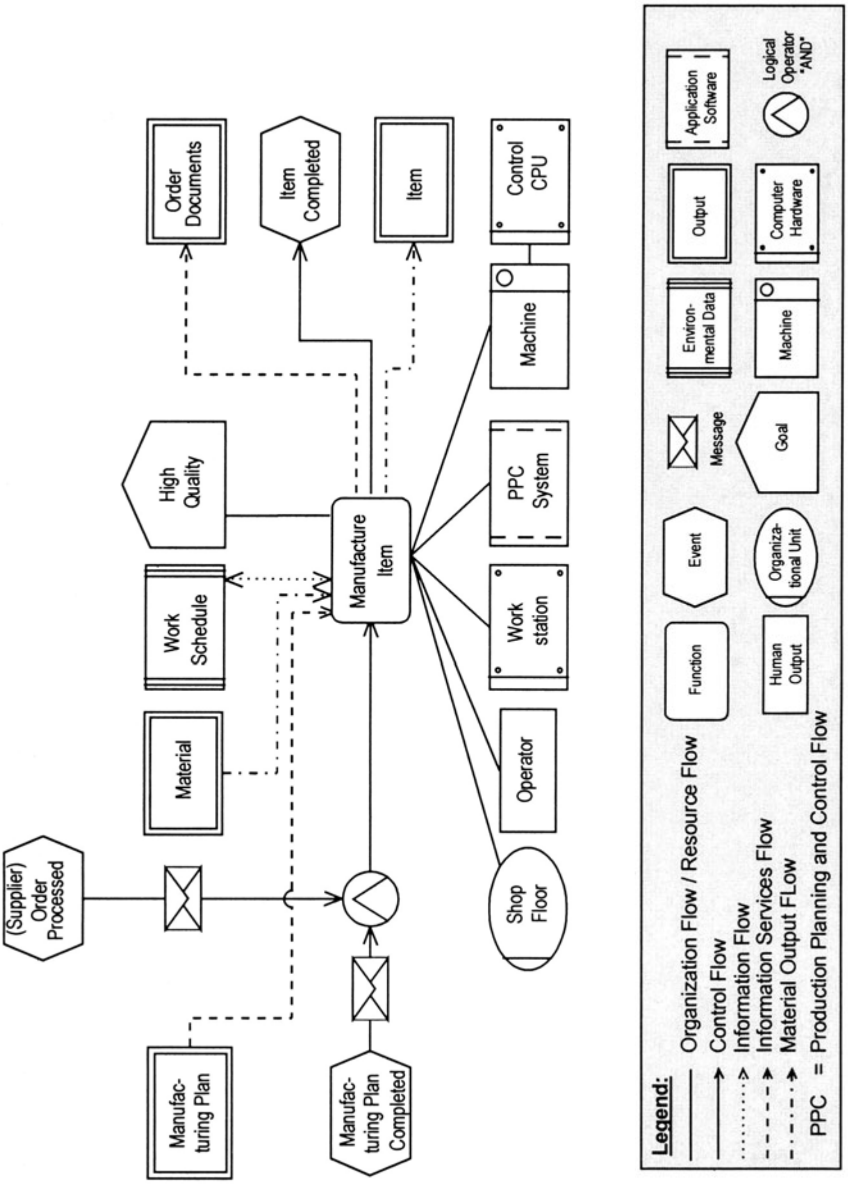


Fig. 7 Detailed excerpt of business process and of event „manufacture item“

Event-driven function flows are also known as event-driven process chains (EPCs). At the Institute for Information Systems (Iwi) at the University of Saarland, Germany, the EPC method was developed in 1992 together with SAP employees in an R&D project financed by SAP AG *(see Keller/Nüttgens/Scheer,*

Semantische Prozeßmodellierung 1992). It is now a key component of the model-supported configuration of the SAP R/3 system. The EPC method does not rigidly distinguish between the flows described here. In particular, output and control flows are frequently consolidated. Messages are not used either. Perhaps it was precisely this simplification that led to its successful real-world application. At a later po int in time, we will expound on the EPC method.

Fig. 7 also depicts in more detail business output creation, characterized by output results by the combination and transformation of the output. Subsequently, the designation "output" will also be used synonymously with the designation "product".

In manufacturing, production factors are combined. According to the production theory as presented by Gutenberg, the basic factors are operating resources, human output, the use of materials, and management. The production theory developed by Gutenberg refers to the creation of material output. However, in this work we need a general approach applicable to the service industry as well, taking into account that service functions are becoming increasingly important in manufacturing industries. Furthermore, in accordance with the information-resource-management concept, information is deemed a production factor in its own right *(see Zimmermann at al., Produktionsfaktor Information 1972; Horton, Information Management Workbook 1981; Krcmar, Informationsmanagement 1997)*.

These requirements are taken into account in modern business production theory, enhancing the concept of production factors (see Fig. 8). Generally, any object required for the execution of the process "production" is regarded as a production factor, regardless of whether it is a material, service or information. The designation "production" is seen in a larger context, including the creation of non-physical output, i.e. services, as well (*see Kern, Industrielle Produktionswirtschaft 1992, p. 12*).

In Fig. 7, the factor "operating resources" is represented by the machine in question, by a computer system (workstation) for production control and by a computer controlling the machine.

Human output or input, respectively, pertaining to the object (object-related output) is represented by linking the qualification "machine operator".

Management -- planning and controlling the combined process -- is enabled by enforcing the goal "high quality", one of the key components of the process execution. Aspects concerning the organizational structure are illustrated by linking organizational units, in this case the shop floor, to the functions.

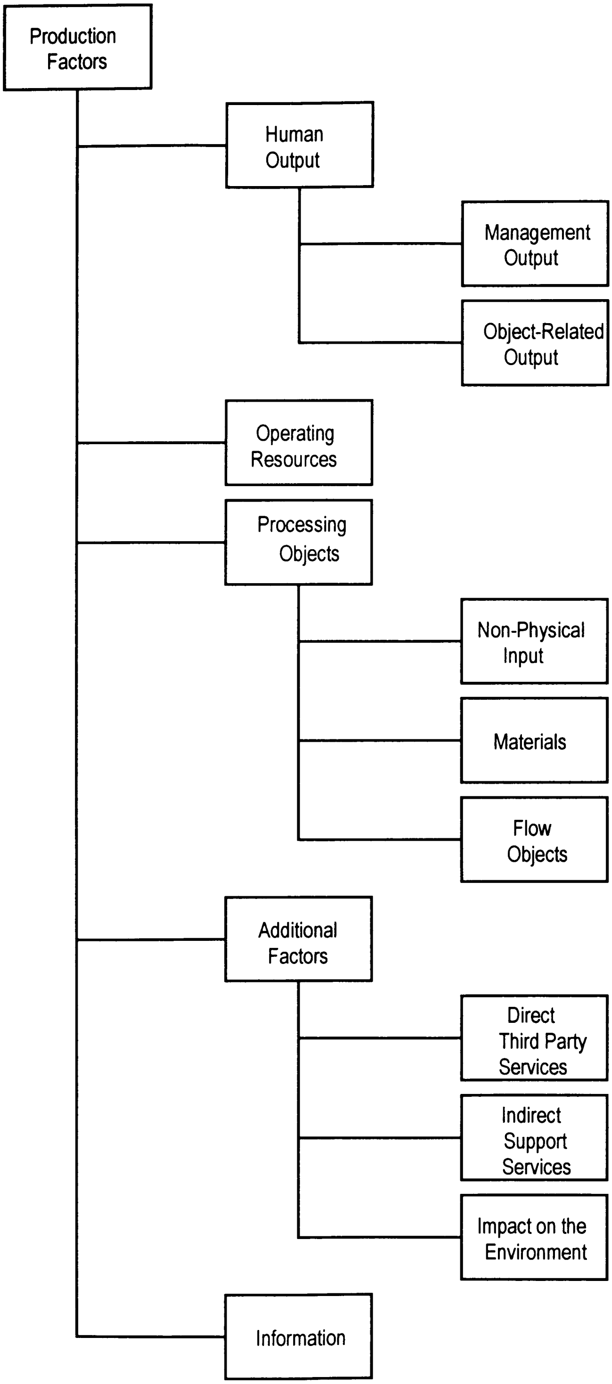


Fig. 8 System illustrating industrial production factors
(according to Kern, Industrielle Produktionswirtschaft 1992, p. 17)

The use of materials as object factors is represented by the respective material. Object factors can also be flow objects, free of charge and provided by the ordering party, e.g. the fabric provided by a dye-works. Other examples of flow objects would be hospital patients, or patrons in a barber shop *(see Corsten, Dienstleistungsproduktion 1994, among other works).*

"Manufacturing plans" are regarded as non-physical materials (services), representing the output of the previous manufacturing planning functions. This is the stage where capacity tests and other checks are carried out. These services are illustrated by the "manufacturing plan" document. In business theory, there is an ongoing discussion as to how to allocate services, as far as production is concerned *(see Farny, Produktions- und Kostentheorie 1965; Müller, Informationsprodukte 1995)*. In this work, we regard them as individual factors, which is the reason they have been enhanced in Fig. 8., compared to their original source.

There are additional factors which, as supporting services, are more indirectly linked with production. This also includes public services and the impact on the environment.

Services provided by external partners refer to production, such as in the case of repair services, but not to the non-physical input directly connected with the processing object, as is the case with our designation "non-physical input".

For the remainder of this work, the simplified classification illustrated in Fig. 9 will be appropriate.

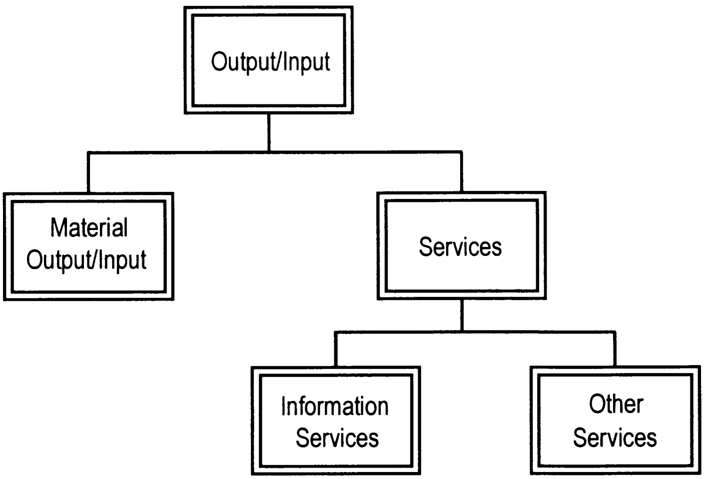


Fig. 9 Classifying types of output/input

Output creating requires additional information about the process. This information is depicted by the data object ”work schedule” in Fig. 7. Work schedules influence the process. Due to the fact that they are not the result of the respective business process, but rather are stored as master data, work schedules cannot be regarded as information services within the process.

In the ARIS business process model, we can distinguish between the events triggering the control flow by means of messages, and the output flow. We should note that, for simplification purposes, control and output flows can be combined in practical modeling. This makes sense when output results are information objects, such as order documents or invoice forms. This makes it possible to equate the triggering event with the information objects. However, for applications requiring a more precise definition of messages, such as work flow systems, or requiring precise tracking of material output flow, such as manufacturing processes, it is absolutely essential to separate flows.

The flows developed in the ARIS business process, are as follows:

- Organization flows:
  Characterize responsibilities and management of organizational units.
- Target flows:
  Characterize business and conceptual goals to be reached by a process or action during execution. Goals are set by management.
- Control flows:
  Control the logical process of functions by means of events and messages. Process functions execute flows by, for example, enhancing input flows by a component, supporting process output yet to be created. In control flows, every process is triggered by one or multiple messages. Each process, however, creates one or more messages as well.
- Output flows:
  We can distinguish between material output flows and service flows. Service flows can operate individually, whereas material output flows are generally controlled and accompanied by service flows. Services are classified into information services with the service consisting of creating and providing information, and other kinds of services. Financial resource flows are components of output flows. To a certain degree, these various services can be substituted. This makes it possible to replace physical services, such as cash, by information services, for example by electronic cash.

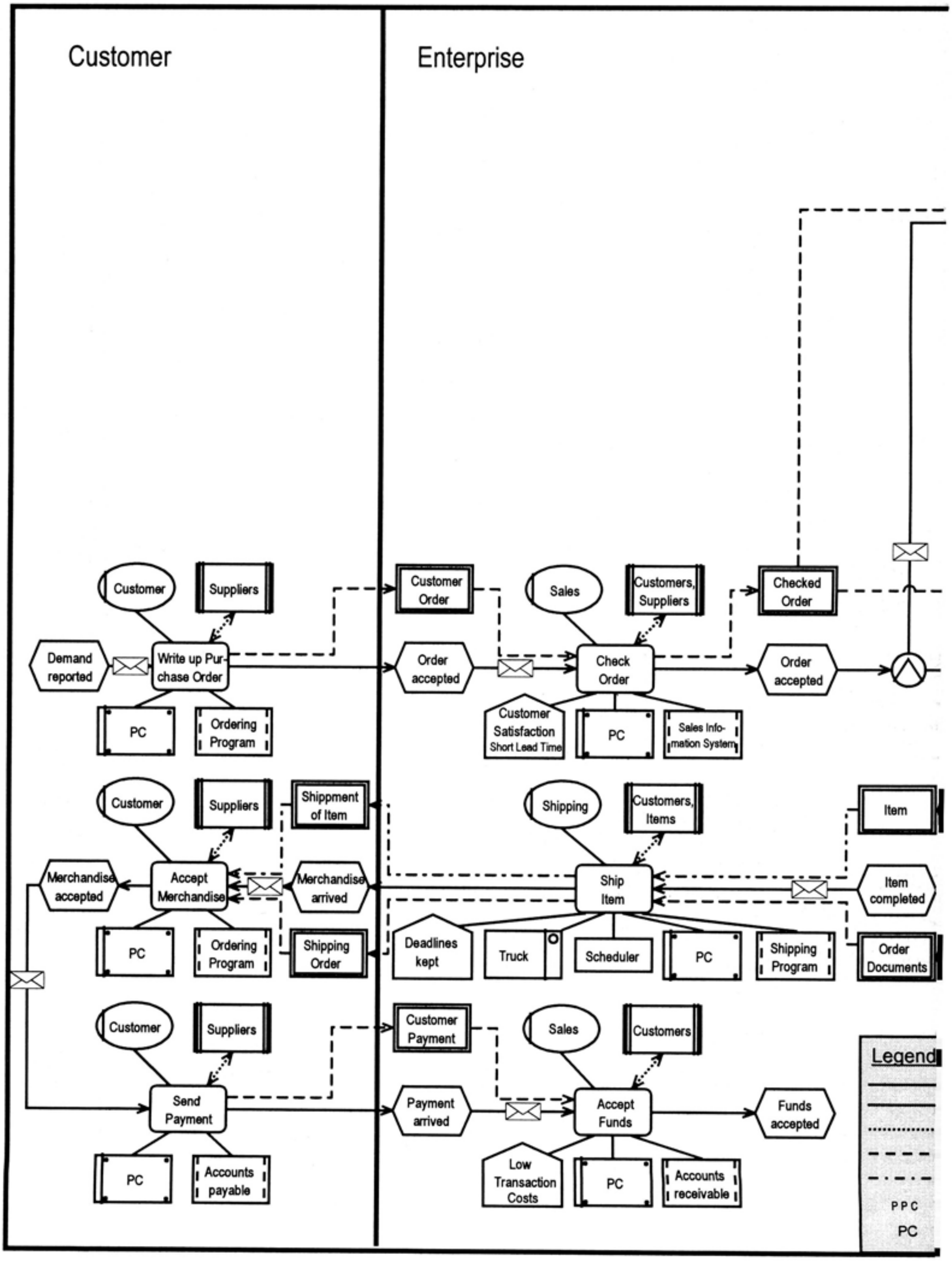
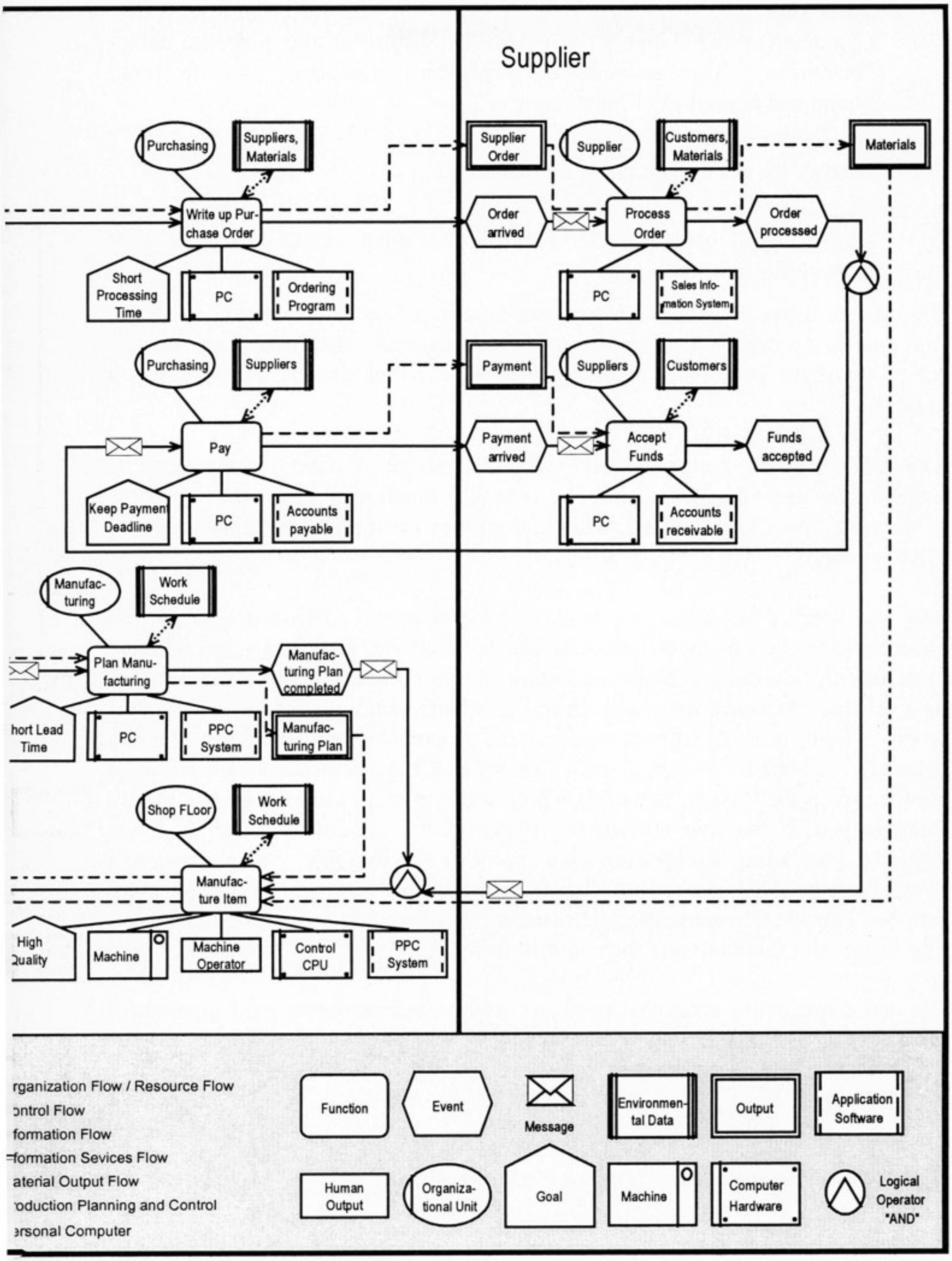


Fig. 10 ARIS business process model


Supplier
Purchasing
Suppliers, Materials
Write up Purchase Order
Short Processing Time
PC
Ordering Program
Supplier Order
Supplier
Customers, Materials
Materials
Order arrived
Process Order
Order processed
PC
Sales Information System
Purchasing
Suppliers
Pay
Keep Payment Deadline
PC
Accounts payable
Payment
Suppliers
Customers
Payment arrived
Accept Funds
Funds accepted
PC
Accounts receivable
Manufacturing
Work Schedule
Plan Manufacturing
Manufacturing Plan completed
hort Lead Time
PC
PPC System
Manufacturing Plan
Shop FLoor
Work Schedule
Manufacture Item
High Quality
Machine
Machine Operator
Control CPU
PPC System
rganization Flow / Resource Flow
ontrol Flow
formation Flow
formation Sevices Flow
aterial Output Flow
oduction Planning and Control
ersonal Computer
Function
Event
Message
Environmental Data
Output
Application Software
Human Output
Organizational Unit
Goal
Machine
Computer Hardware
Logical Operator "AND"

- Resource flows:
  Characterize the delivery of utilization output of the potential factor "resources". The designation "resources" comprises manufacturing equipment as well as IT hardware.
- Human output flows:
  Display the delivery of direct human output.
- Information flows:
  Consisting of goal-oriented skills for the execution of functions, they control information access.

Fig. 10 describes the complete business process of order processing in detail. Shipping as a transport function is displayed as material output because it changes the place of the processing object, However, it could also be interpreted as a service.

The ARIS business process model is hierarchical, i.e., functions can in turn be explained by more detailed business processes. A function is regarded as a process at the next lower level as well, when the process can be described by the same ARIS elements.

The objective of the respective display elements in the ARIS business process model is to create a method-independent view. However, in some business-related system analysis concepts, display elements of object oriented analysis are depicted as a suitable descriptive language. In object oriented analysis, however, the focus is on defining classes, sub-classes and the respective methods. These classes generally correspond to data classes. Control flows are depicted by the message flow between the classes. In business processes, orders, customers and items are interchanged in multiple messages. Display of the control flow can be quite cluttered when using an object oriented approach. For example, the same message flows must be distinguished from one another by the selection of a subsequent number. However, because the ARIS concept is focused on business concepts, the key focus is on functions and their control flows.

We will subsequently show that an object oriented approach is a good supplement to process-oriented modeling because it enables an additional view of the business process models. Object oriented modeling can therefore be integrated into the ARIS concept.

B.II.2 The Generalized Business Process Model

Business process models can be designed at various abstraction levels. The previously discussed business process model were based on an order processing application. This example did not describe the real-world procedure of customer orders, but rather the general order processing process which is an abstraction of the realized process. This kind of description is known as a business process type.

Fig. 11 illustrates an excerpt of the manufacturing process of an individual order processing process. Here, every object involved in the business process is instantiated by the affixed name or names. Individual business process models are used for controlling individual processes. In manufacturing, this is customarily carried out by creating work schedules as the manufacturing process descriptions for individual parts or manufacturing orders.

In office management, individual business process models are executed through workflow control systems. Workflow systems control document flows and work flows electronically. Therefore, they must have access to information regarding the respective control structure and responsible entities or devices for every single business event. Individual business processes are known as instances. There is a class - instance relationship between the business process type in Fig. 7 and the process instance in Fig. 11.

All individual order processes make up the class or type "order processing business process". The individual processes are instances (elements) of this class. Classes take on the characteristics of the elements, although the individual instances are abstracted.

Type levels are the most important levels in business process modeling. In order to support (re)organization measures, not only is know-how regarding each business process necessary, but also know-how regarding the entire general process structure. After all, the organizational changes are carried out in order to improve the process as a whole. Instances thus proceed according to the new schema. Due to exception handling regarding the process structure, individual variances of the instances can be taken into account.

The illustration of instances is known as description level 1; type levels are known as description level 2.

Levels 1 and 2 thus have the same relationship as classes and instances. Every class is characterized by its name and the enumeration of its attributes, by which the instance is described. For example, the class CUSTOMER is characterized by the attributes "customer number", "customer name" and "payment period". The instances of these characteristics are the focus of the descriptions at level 1. Fig. 12 depicts a few examples of levels 1 and 2.

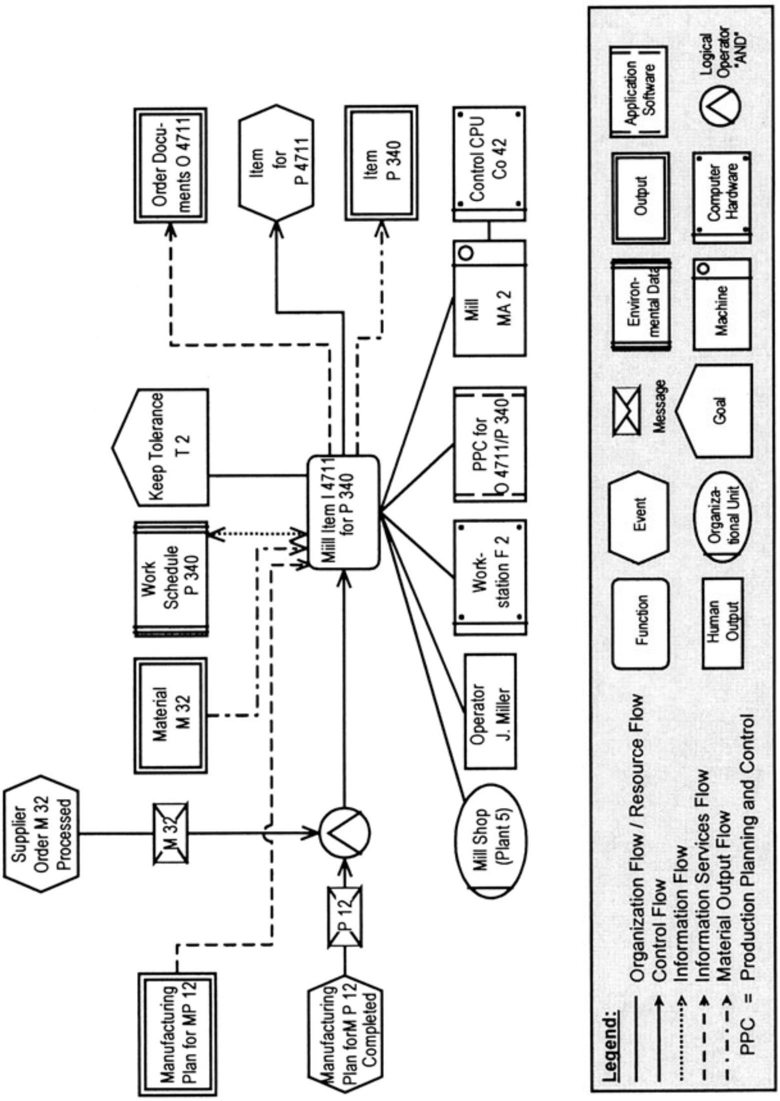


Fig. 11 Business process model of an order instance (level 1)

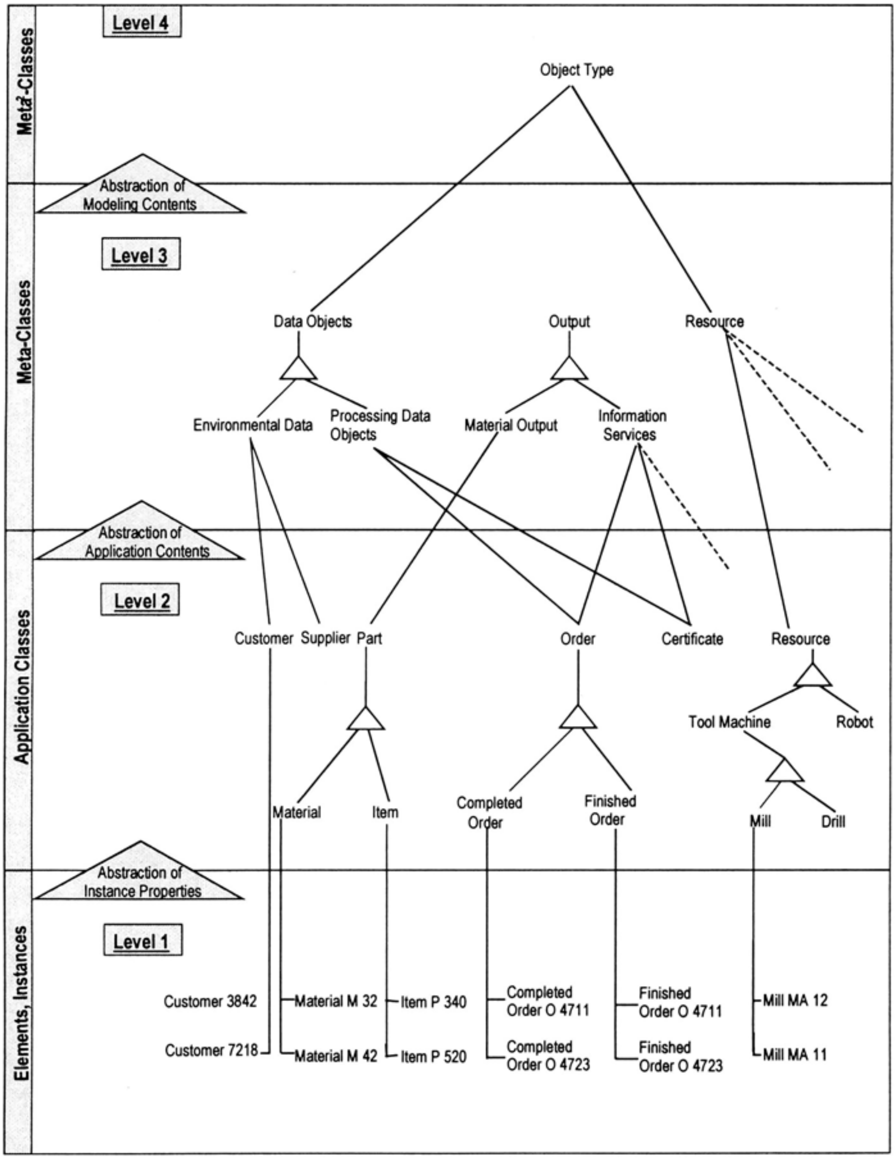


Fig. 12 Abstraction levels in modeling

In order to further characterize classes, we can list the functions applied to them. We will do this at a later point in time.

Grouping classes is always a slightly task. Therefore, when defining order designations, we will only abstract specific properties of cases 4711 or 4723, respectively, leading to the classes "completed order" or "finished order". At level

2, we will abstract the "completed" and "finished" properties and create the parent class "order" from the subset. This operation is known as generalization and is illustrated by a triangular symbol.

When quantities are generalized, they are grouped to parent quantities. This makes order instances of level 1 instances of the class "order" as well. The class "order" is designated as the property "order status", making it possible to allocate the process state "completed" or "finished" to every instance. Materials and items are also generalized, making them "parts" and "resources".

Thus, level 2 contains application-related classes of business process descriptions. On the other hand, with new classes created from similar classes of level 2 by abstracting their application relationships, these are allocated to level 3, the meta level, as illustrated in Fig. 12. Level 2 classes then become instances of these meta classes. For example, the class "material output" contains the instances "material" and "item" as well as the generalized designation "part". The class "information services" contains the class designation "order", along with its two child designations, and the class designation "certificate". The creation of this class is also a function of its purpose. Thus, either the generalized classes of level 2 or their subclasses can be included as elements of the meta classes.

When creating classes, overlapping does not have to be avoided at all costs. For example, from an output flow point of view, it is possible to create the class "information services" from the classes "order" and "certificate". Conversely, from the data point of view, these are also data objects, making them instances of the class "data objects" as well.

If this procedure is applied to the business process model in Fig. 10, described at level 2, this leads to the general ARIS business process model at level 3, depicted in Fig. 13. This figure contains the general description classes of business processes, along with their relationships. The relationships depicted by arrows could also be expressed as classes (relationship classes). For simplification, this has been avoided. When we subsequently speak of meta classes, we mean every representation object (classes and relationships).

In addition to the relationships depicted here, other relevant relationships between the classes are feasible. It is also possible to create subclasses from the classes of the meta level. The model in Fig. 13 shows the essential objects necessary for illustrating business processes, although this figure is not necessarily complete.

Thus, the classes at modeling level 3 define every object necessary for describing the facts at level 2. These objects make up the building blocks for describing the applications at level 2. On the other hand, because the classes at level 2 comprise the terminology at level 1, objects at level 3 are also the framework for describing the individual business processes.

This abstraction process can be continued by once again grouping the classes at level 3 into classes which are then allocated to the meta^2 level . Next, the content-related modeling views are abstracted. In Fig. 12, the general class "object type" is created, containing all the meta classes as instances.

We will discuss the modeling levels, in particular a more specific description of the meta^2 level, in more detail in the chapter "modeling levels".

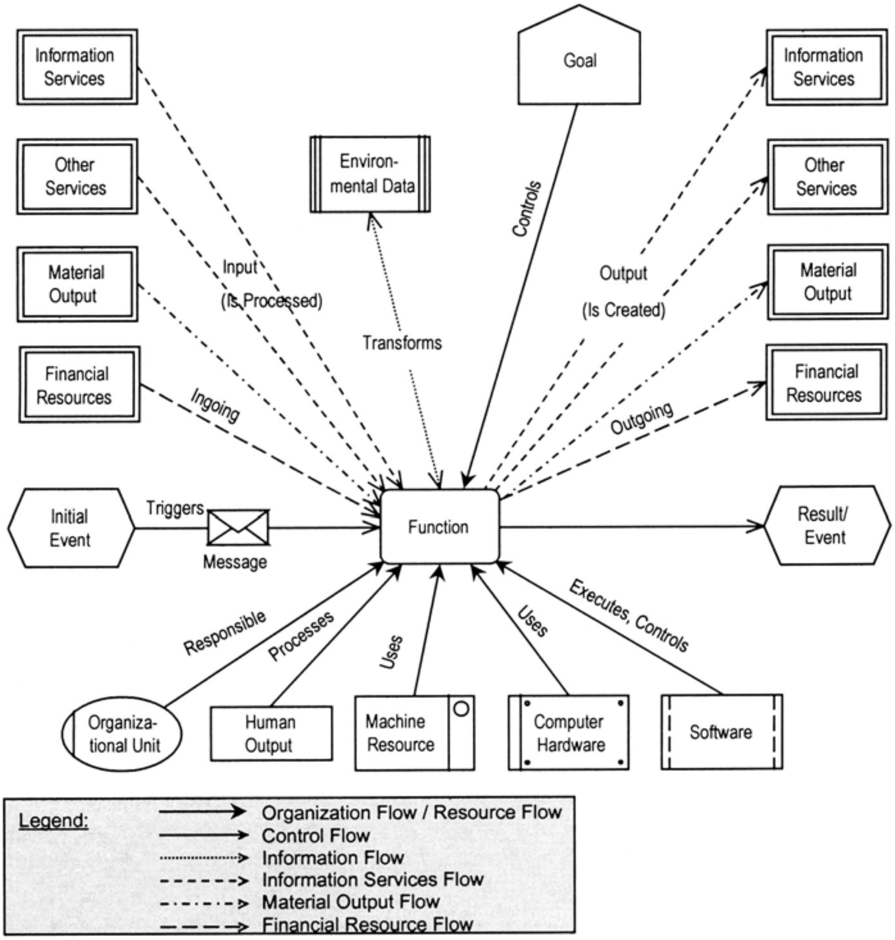


Fig. 13 The general ARIS business process model

C Developing the Architecture of Integrated Information Systems (ARIS House)

The meta business process model at level 3 defines designation classes and interrelationships, with which actual business application processes of levels 2 and 1 can be modeled. Due to the fact that the objects at level 2 are instances of the meta level, only those objects for which classes are defined at the meta level can be utilized. Conversely, the business process types modeled at level 2 determine the structure of the actual individual procedures at level 1. Thus, meta models basically determine the capability of business process design.

Due to the wide variety of classes and their semantic interrelationships, business process models are structured in greater detail. The resulting structure is known as the "Architecture of Integrated Information Systems" (ARIS).

The classes contained in the meta business process model in Fig. 13 are as follows:

- Environmental data of the process,
- Initial and result events,
- Messages,
- Functions,
- Human output,
- Machine resources and computer hardware,
- Application software,
- Material output, service output and information services,
- Financial resources,
- Organizational units,
- Corporate goals.

Because every class can be linked with another, the system structure is complex. The semantic relationships between the classes and the respective function class, illustrated in Fig. 13, only demonstrates an excerpt of all possible interrelationships.

Multiple relationships are also possible among the classes. For example, the relationship between functions and human output also depends on which resource supports the execution of the process. Furthermore, relationships within the classes can describe how data objects depend on one another or how events are linked with one another.

In order to reduce complexity, in chapter C.I, classes with similar semantic interrelationships are grouped into ARIS views. This makes it possible to view

coherences within a view, without immediately having to take the interrelationships with other views into account.

Up to now, we have treated business processes more from a business administration point of view, without focusing on the use of IT in detail. However, since the implementation of business models in information systems is a key part of this work, we will introduce a life cycle concept in chapter C.II, transforming business process classes into IS objects step by step.

In order to describe these interrelationships in a more detailed manner, a more conceptual description language than in the previous illustrations is necessary. This language is developed in chapter C.III, for creating an information model draft. The ARIS concept also illustrates how to describe business processes. For this purpose, we will develop a procedural model draft in chapter C.IV.

C.I ARIS Views

The grouping of classes and their relationships into views serves the purpose of structuring and streamlining business process models. Splitting up views has the added advantage of avoiding redundancies which can occur when objects in a process model are used more than once. For example, the same environmental data, events or organizational units might be applied to several functions. View-specific modeling methods which have proven to be successful can also be used. Particularly in this light, view procedures differ from the more theoretical modeling concepts, where systems are divided into subsystems for the purpose of reducing complexity. In principle, however, every subsystem is depicted in the same way as the original system. This is why it is not possible to use various modeling methods in the same system.

ARIS views are created according to the "semantic correlation similarity" criterion. They are shown in Fig. 14a-d, based on the meta business process model shown in Fig. 13. Due to the direct link of the description levels, the views shown in the meta model are also valid for levels 2 and 1.

Although only preliminary class designations are used in meta business process models, and these primarily refer to business activities, these views will at a later point in time also be suitable for detailing and implementing classes into IT designations of the life cycle concept.

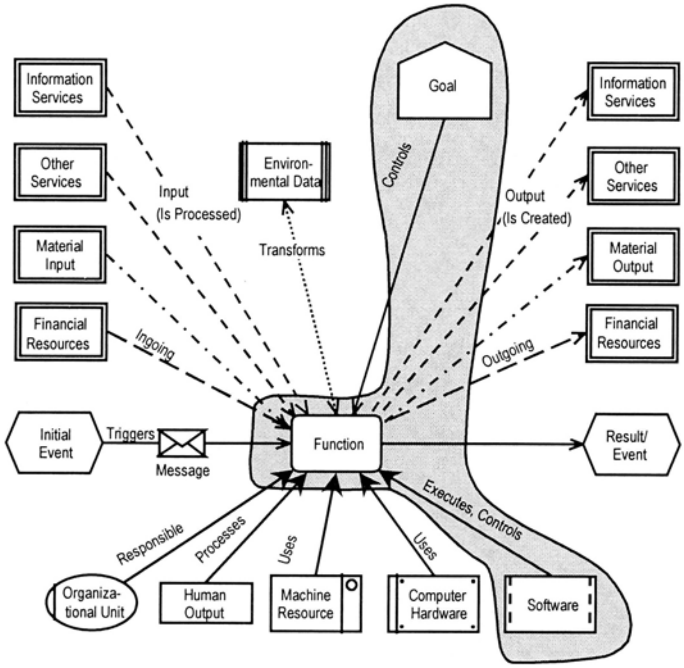


Fig. 14a Function view

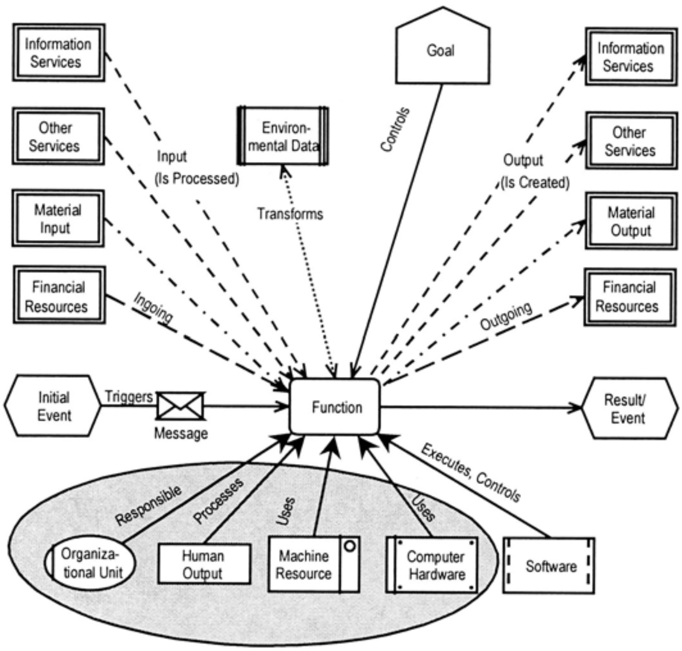


Fig. 14b (Hierarchical) organization view

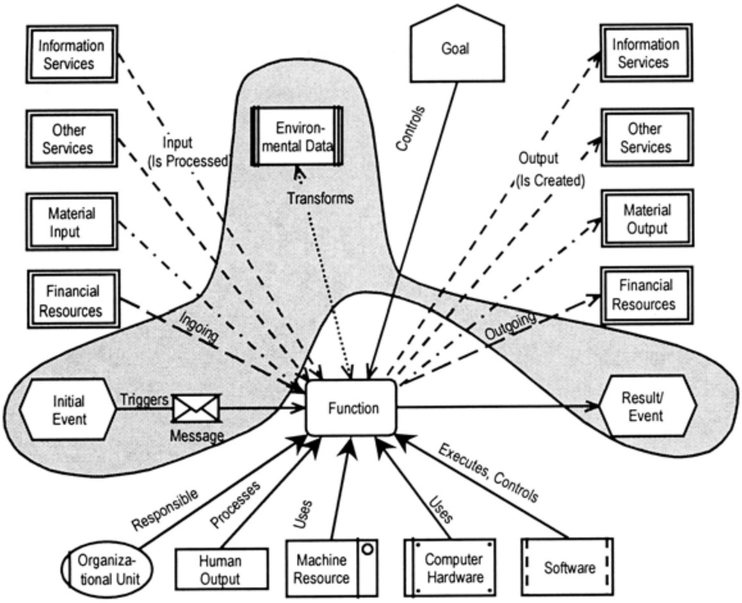


Fig. 14c Data view

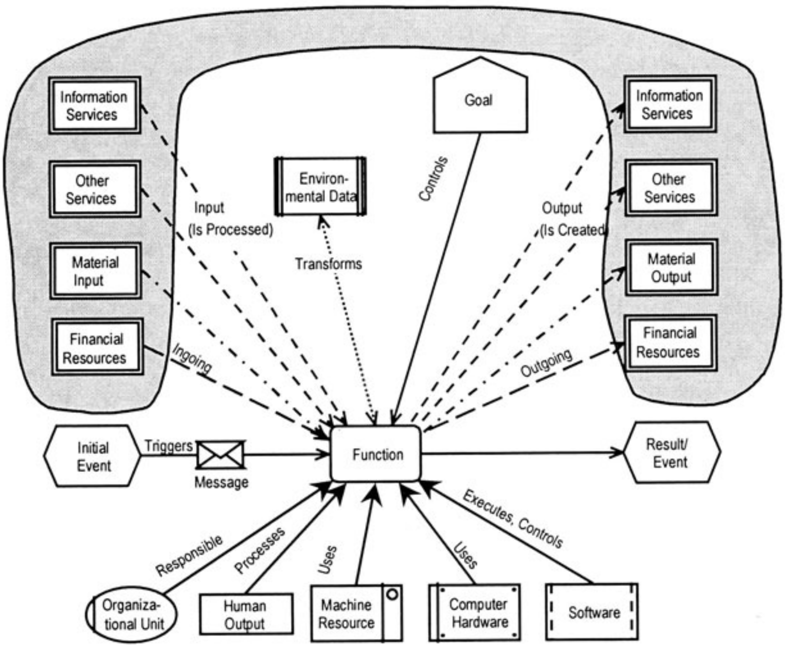


Fig. 14d Output view

When creating views, the various flows developed in B.II.1 are utilized, leading to the definition of the following ARIS views:

- **Function views**

  The processes transforming input into output are grouped in a function view in Fig. 14a. The designations "function", "process" and "activity" are used synonymously.

  Due to the fact that functions support goals, yet are controlled by them as well, goals are also allocated to function views -- because of the close linkage. In application software, computer-aided processing rules of a function are defined. Thus, application software is closely aligned with "functions", and is also allocated to function views.

- **Organization views**

  The class of organization views creates the hierarchical organization structure, also known as the organization view. Organization views are created in order to group responsible entities or devices executing the same work object. This is why the responsible entities "human output", responsible devices, "financial resources" and "computer hardware" are allocated to the organization view (see Fig. 14b).

- **Data views**

  Data views comprise the data processing environment as well as the messages triggering functions or being triggered by functions. Preliminary details on the function of information systems as data media can also be allocated to the data names. Information services objects are also implicitly captured in data views. However, they are primarily defined in the output view. Fig. 14c illustrates data view objects.

- **Output views**

  Output views contain all physical and non-physical input and output, including funds flows (see Fig. 14d).

- **Control views / Process views**

  The views are where the respective classes with their view-internal relationships are modeled. Relationships among the views as well as the entire business process are documented in the control or process views, creating a framework for the systematic inspection of all bilateral relationships of the views and the complete process description.

In system theory, we can distinguish between the structure and the behavior of a system. Structures encompass the static view of a system, whereas behavior describes its dynamics. In business process models, dynamics are expressed by event control and message flow. The function, organization, data and output views, respectively, describe the system structure. Control views show all the structural coherences of the views and the dynamic behavioral aspect of the business process flow.

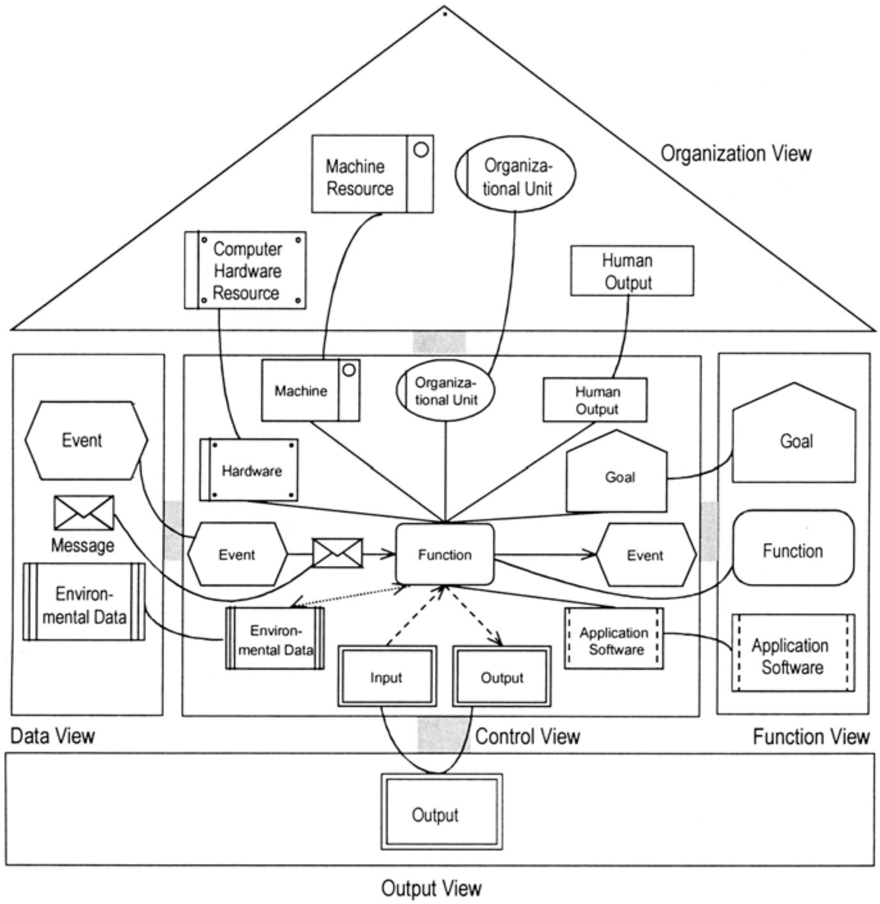


Fig. 15 Views of the ARIS house

Fig. 15 depicts the views and some of their classes, resulting in the ARIS house. The process model is shown in the control view. The origin of the control view's objects as well as individual views are also illustrated.

C.II ARIS Phase Model

Up to now, we have discussed business processes from a management point of view, i.e., without any particular focus on information technology. The aforementioned application programs (components of the function view), computer hardware (a component of the organization view) and data media (components of the data view) only contain system names, not IT descriptions. These are included in the ARIS concept by evaluating the IT support provided by each ARIS view.

- Function views are supported by the application programs which may be described in more detail by module concepts, transactions or programming languages.
- Organization views, along with their production resources and the computer resources responsible, may be detailed further by listing network concepts, hardware components and other technical specifications.
- Data views may be detailed more precisely by listing data models, access paths and memory usage.
- Output views group the various types of output, such as material output and information services. Here again, there is a close alignment with the supporting information technology. In material output (such as entertainment technology, automobiles and machine tools) more and more IT components (for example, chip technology), along with the necessary hardware, are used. Other service industries, such as airline reservations, are closely linked with IT as well.
- Due to the fact that the respective views can be combined within the control view, there is a definite link with IT, as demonstrated by the above arguments.

Using a phase model, business descriptions are thus transformed step by step into information and communication technology objects.

Phase models characterize the description steps for implementing business issues by means of computer systems. This implementation process is often described by different concepts *(see Olle et al., Information Systems Methodologies 1991, pp. 46 or Gesamtstruktur des V-Modells, in Bröhl/Dröschel, V-Modell 1995, pp. 21-30)*. In the ARIS model, five different steps are used (see Fig. 16).

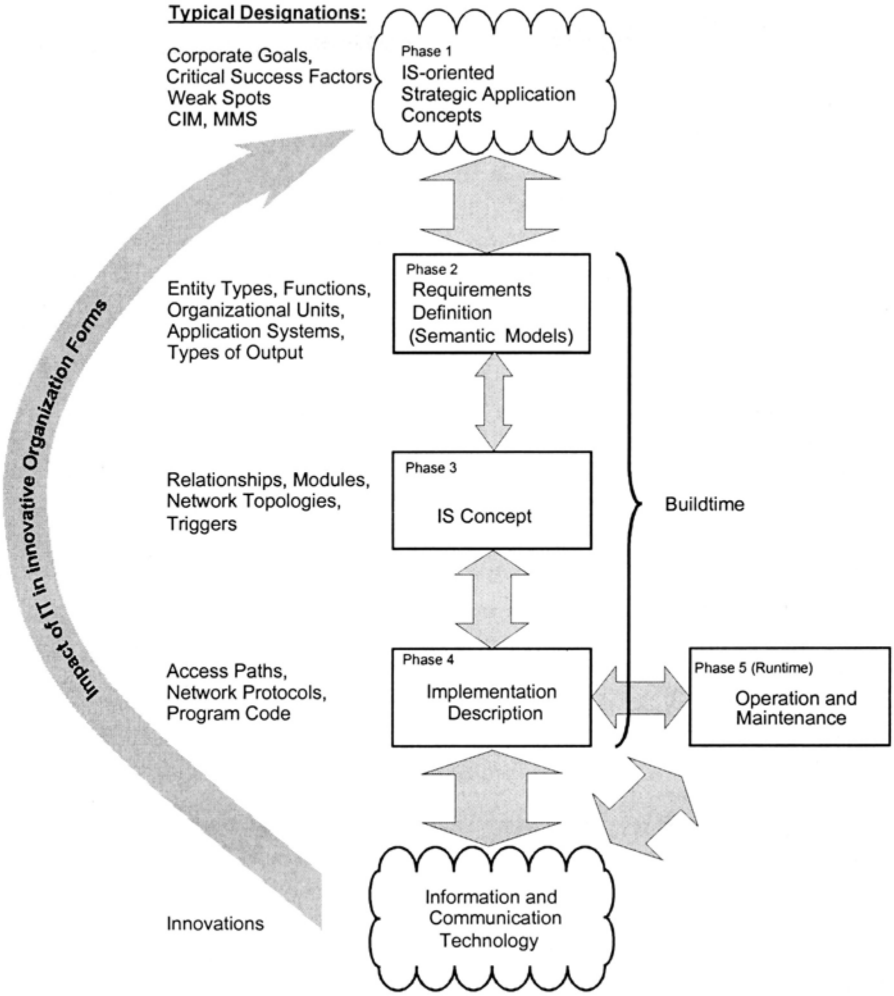


Fig. 16 ARIS phase model

In phase 1, the IS-oriented **initial strategic situation** is established. "IS-oriented" means that basic IT effects on the new enterprise concepts are already taken into account. Some examples of these relationships would be creating virtual companies through communication networks, PC banking, integrated order processing and product development in industry (CIM) or integrated merchandise management systems (MMS) in retail.

Strategic corporate planning determines long-term corporate goals, general corporate activities and resources. Thus, planning has an effect on the long-term definition of business processes, influencing corporate goals, critical success

factors and resource allocation. The methods in question are similar to management concepts for strategic corporate planning. Provided actual business processes have already been described, this occurs in a general fashion. At this stage, it is not advisable to split up functions into ARIS views and then describe them in detail.

In Phase 2, the **requirements definition,** individual views of the application system are modeled in detail. Here as well, business-organizational content is key. Examples for business processes, especially the ARIS business process model discussed in Fig. 10, should be included at this level.

However, in this phase more conceptual description languages should be used than in the strategic approach, due to the fact that the descriptions for the requirements definition are the starting point for IT implementation. Description languages that are understandable from a business point of view should be used, yet they should be sufficiently conceptual to be a suitable starting point for a consistent IT implementation. Therefore, it makes sense to include general IT objects, such as databases or programs, at this level.

Phase 3 calls for creating the **design specification,** where business models are adapted to the requirements of the implementation tool interfaces (database, network architectures or programming languages, etc.). At this time, actual IT products are still irrelevant.

Phase 4 involves the **implementation description,** where the requirements are implemented in physical data structures, hardware components and real-world IT products.

These four phases describe the creation of an information system and are therefore known as "buildtime". Subsequently, the completed system becomes operable, meaning it is followed by an operations phase, known as "runtime". This work does not address the operation of information systems, i.e., runtime, in great detail.

The requirements definition is closely aligned with the strategic planning level, illustrated by the width of the arrow in Fig. 16. However, it is generally independent of the implementation point of view, as depicted in the narrow arrow pointing to the design specification.

Implementation description and operations, on the other hand, are closely linked with the "IT equipment and product level". Changes in the system's IT have an immediate effect on its type of implementation and operation.

The phase concept does not imply that there is a rigid sequence in the development process, as in the "waterfall model". Rather, the phase concept also

includes an evolutionary prototyping procedure *(see Boehm, Spiral Model 1988 und Floyd, Systematic Look at Prototyping 1984, among other works)*. However, even in evolutionary software development, the following description levels are generally used. Phase models are primarily used because of the fact that they offer a variety of description objects and methods.

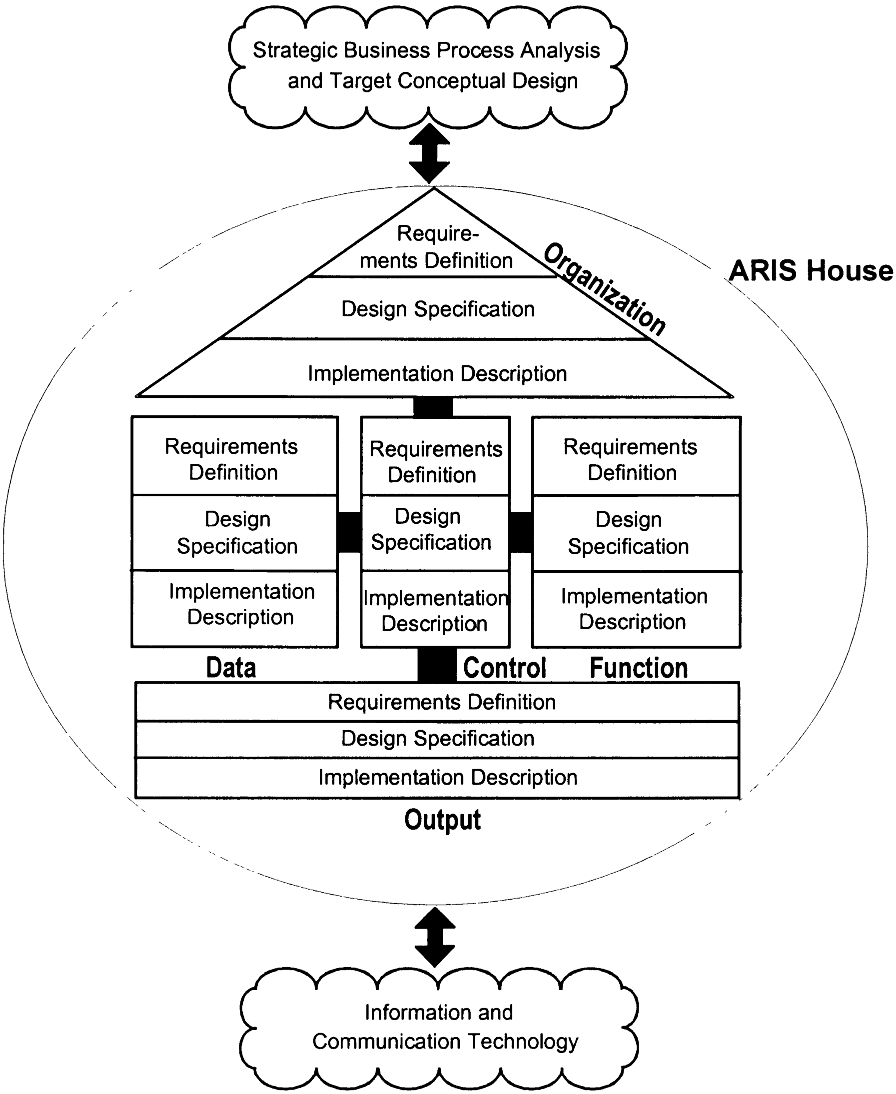


Fig. 17 ARIS house with phase concept

The ARIS house in Fig. 17 is enhanced by the four phases of the buildtime ARIS phase model. After a general conceptual design, the business processes are divided into ARIS views, and documented and modeled from the requirements definition to the implementation description. These three description levels are created for controlling purposes as well. This makes it possible to create the links to the other components at each of the description levels.

The ARIS house in Fig. 17 illustrates the architecture of an information system. Comprised of the description views, this architecture is divided into the requirements definition, design specification and implementation description, respectively, and is closely related to IT issues.

The ARIS concept aims at creating and controlling operational business processes. In addition to its link with strategic corporate planning, it is also aligned with strategic information management (*regarding information management, see Krcmar, Informationsmanagement 1997; Schmidt, Informationsmanagement 1996, among other works*).

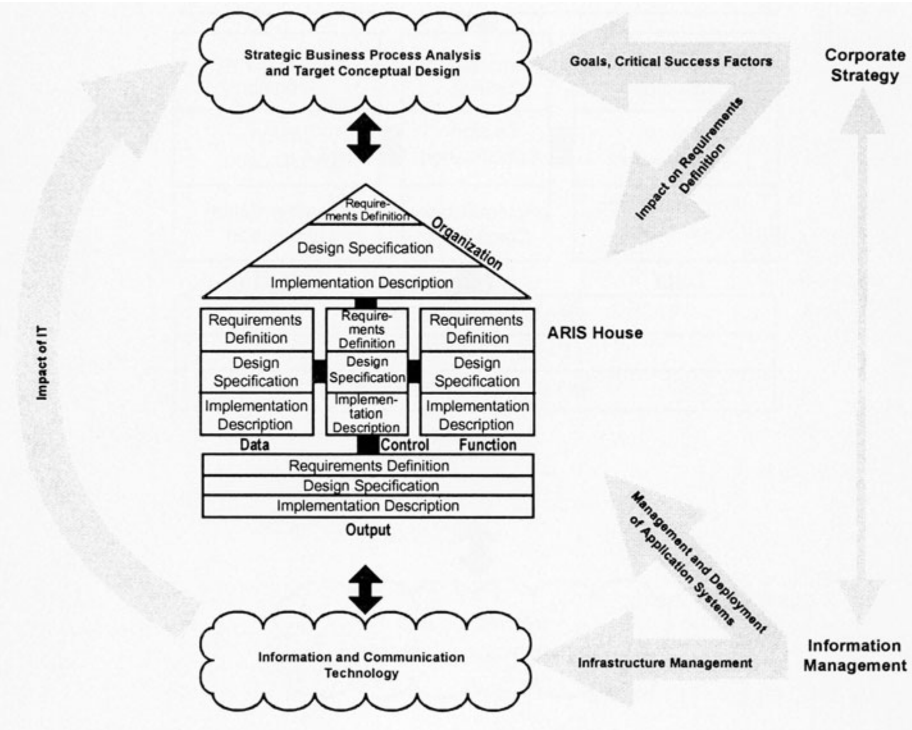


Fig. 18 ARIS house with links to corporate strategy and information management

Information management is concerned with planning, controlling, and deploying the "information" resource. In a reference model (*see Wollnick, Referenzmodell des Informationsmanagements 1988*), Wollnick divides these tasks into three areas: infrastructure management (information technology management), application system management and managing the deployment of information systems. These definitions can also be adapted for the ARIS concept. The first task of the technology infrastructure refers to information technology and the design specification levels depicted in Fig. 18. The second task refers to management of the information systems and is characterized by the implementation of organizational requirements definitions into computer-aided information systems by means of the ARIS life cycle concept, the main focus of the buildtime view of the ARIS concept. The third task, managing the deployment of information systems, refers to the runtime phase of the ARIS life cycle model.

Due to the fact that information technology impacts organizational challenges and the resulting changes (see left arrow in Fig. 18), there is also a link between information management and the enterprise strategy, depicted on the right hand side of the illustration. Thus, the ARIS concept can help make the implementation of strategic enterprise concepts more transparent, also creating a framework for a better implementation of the actual purpose of information management.

C.III Preliminary ARIS Information Model

The ARIS house establishes a framework for classifying the descriptive components of a business process. We will now take a closer look at the individual building blocks of the business process, along with their relationships. We will study the meta level where the elements of a general business process, i.e., without any context toward a particular business process, are captured. This is based on the ARIS business process model, depicted in Fig. 13, which we will now discuss in greater detail by examining the relationships among the elements. For this purpose, we will use a unified description language and unified symbols for the various elements, such as functions, organizational units, resources, messages, etc., as well as for the relationships among them.

Chen's entity relationship model (ERM) (*see Chen, Entity-Relationship Model 1976*) is well suited for depicting objects. Although it was originally designed to illustrate data structures in application systems, it can also be used as a general description language and, thus, for describing meta levels.

For object oriented approaches as well *(according to Rumbaugh et al., Object-Oriented Modeling and Design, 1991*), classes and their relationships are entered in the object model. However, methods are also allocated to them. Regarding

modeling objects, at the meta level they are identical (such as when creating, deleting, editing or graphically displaying an object).

With object oriented methods, it is also permissible to only use classes and the respective names during the analysis phase, i.e., skipping attributes and methods *(see detailed discussion on modeling methods and meta models in Scheer, ARIS - Business Process Modeling 1998).*

In the first edition of this book, we used an enhanced ERM as a description language. However, since the unified modeling language (UML); *see UML Notation Guide 1997; UML Semantics 1997; for deployment of UML, see Oestereich, Objektorientierte Softwareentwicklung 1997, among other works*) is gaining approval as a standard object oriented modeling language, we will use this language's illustration of classes. As far as content is concerned, both languages are interchangeable.

The UML description language makes it possible to individually model object classes and association classes, respectively, in the various views. This description is known as the **ARIS meta model** or **ARIS information model.**

At the same time, this information model describes the database design, where real-world models developed according to the ARIS methodology can be stored. Organization, function, data, output and control models, respectively, of an application are regarded as instances of the database, designed according to the information model. These databases are known as **repositories**. The designation "repository" became popular around 1989, when IBM announced its AD/CYCLE software development concept *(see Winter/Maag, AD/CYCLE 1990; Fosdick, Ten Steps to AD/CYCLE 1990; Hofinger, IBM Repository Manager 1991, among other works).*

For every ARIS view (function, organization, data, output and control, respectively), the ARIS repository contains models of description level 2 as well as their relationships and models for each phase of the ARIS life cycle. When modeling is carried out at level 1, i.e., at the ARIS instance level 1, the repository must be updated by storing the process instances.

Thus, repositories become the core of the information system. The importance of the information model within the repository becomes obvious because of its ability to determine the efficacy of the description elements.

The objects used by UML are class diagrams, modeled by boxes and associations, in turn depicted by borders. Associations are distinguished by 1:*, 1:1, *:* or *:1 cardinalities. The asterisk stands for "many" or "n".

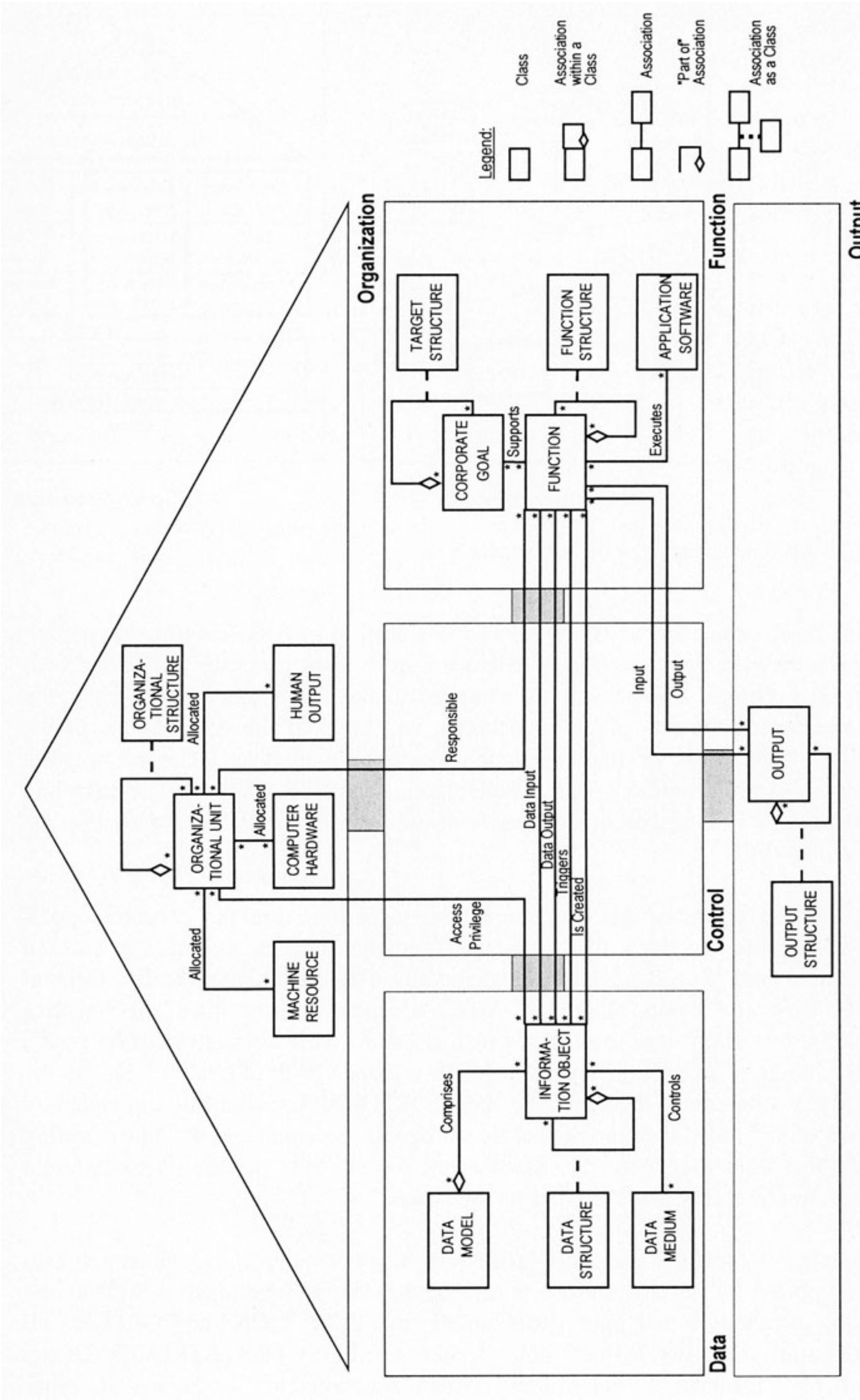


Fig. 19 Preliminary ARIS information model

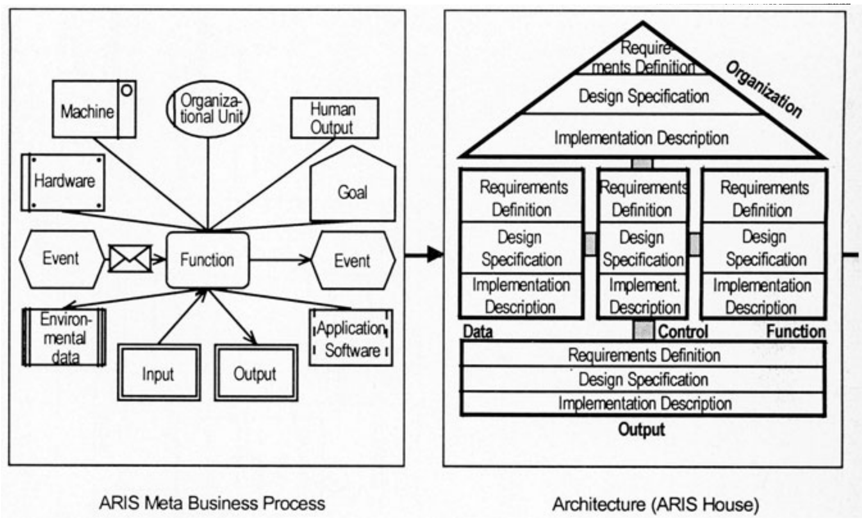
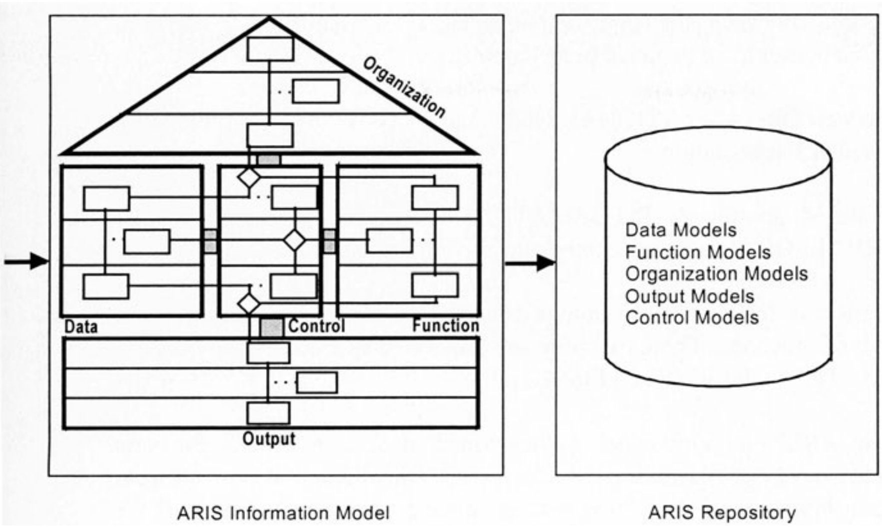


Fig. 20 ARIS components of the ARIS meta level

Using these simple elements, Fig. 19 shows a draft of an ARIS information model. In each view, only a few classes discussed up to now are described, along with their associations. Of the various elements in the life cycle concept, only the requirements definition phase is included, i.e., none of the designations of the design specification or implementation description are used. The information model in Fig.19 provides a short introduction of this type of model. The modeling methodology is described in much more detail in *Scheer, ARIS - Business Process Modeling 1998*.

The starting points of the function model in Fig. 19 are the corporate goals controlling the functions; that is, certain functions must be executed to reach a particular goal. Corporate goals are generally classified hierarchically. General goals, such as "maximize profit", "reach a certain market share" or "reach a certain growth rate" lead to subgoals such as "attain a certain amount of revenue", "lower costs by a certain amount" or "reach a certain level of quality". Due to the integrated structure of the goals, the class CORPORATE GOALS is associated by means of *:* links. Due to the fact that subgoals are contained in the overriding goals, this leads to a "part of" - association. We call this "part of"- association a target structure. It is maintained as its own class.

Examples of functions are order processing, sales or controlling, which can also be supported by derived subfunctions. The linking of functions, as well as the linking of functions to support goals, implies that FUNCTION and CORPORATE GOALS are associated by means of *:* links. The FUNCTION STRUCTURE is a "part of"- association, determining which functions are contained in other functions.



The central designation in the organization view is ORGANIZATIONAL UNIT. This class has instances such as POSITION, DEPARTMENT or the ENTERPRISE. Regardless of whether these areas are subordinate or overriding, both lead to an *:* - "part of" association within the class ORGANIZATIONAL UNIT. Thus, the association makes it possible for one area to be subordinate to several other areas. This is the case when, for example, a sales department is responsible for several overriding product areas. The responsible entities or devices, respectively, "machine", "computer" and "human output" are allocated to organizational units.

On the left side of the ARIS house, the data view depicts the data structure model. The class INFORMATION OBJECT characterizes objects to be described by database attributes. There are associations (for example, which customer buys which items) between their instances, such as item data and customer data. These associations are expressed by an *:* - association within the class INFORMATION OBJECT.

Information objects of an area with correlated content can be grouped in a class diagram or a data model. Because they can overlap due to identical information objects, DATA MODEL and INFORMATION OBJECT are linked by means of *:*- "part of" associations.

In the output view, the class OUTPUT represents every kind of output (material output, service and information output). The instances in question would be application-related output classes such as item, materials, spare parts, assembly hours, warranty services or certificates. Here as well, various kinds of output can be linked with one another (by "part of" associations).

The associations between the four components, i.e., organization, function, information, and output, are depicted in the control view.

The link between ORGANIZATIONAL UNIT and FUNCTION is expressed by the RESPONSIBLE association.

Certain privileges relating to INFORMATION OBJECTS, expressed by the ACCESS PRIVILEGE association, can be allocated to organizational units.

Functions transform input data into output data. Events trigger functions and are also the result of functions. These functions are illustrated as associations between the INFORMATION OBJECTS and FUNCTIONS.

The complete ARIS function model, as developed in *Scheer, ARIS - Business Process Modeling 1998,* describes the classes and the meta business process model relationships among them in a detailed manner. It also describes all the ARIS views, across the life cycle phases. The model consists of approximately 300 classes and associations.

The ARIS information model is the repository design scheme for storing the respective application models. The stored repository data contain the classes of real-world applications, such as in sales or accounting, although usually at the type level. Whereas designations such as CUSTOMER and ITEM are stored as instances of the class INFORMATION OBJECT within the repository. The instances, i.e., the individual customer and item entities of level 1, are generally stored in the sales database. When modeling instance processes, such as for workflow applications, this rule does not necessarily have to be observed.

Fig. 20 groups the four components of the meta level,

- ARIS meta business process,
- Architecture (ARIS house),
- ARIS information model,
- ARIS repository

along with their interrelationships.

C.IV Preliminary ARIS Procedural Model

The ARIS concept provides a methodology for describing business processes, from the requirements definition to the implementation description. Order processing, complaints processing or insurance claims processing are examples.

However, the execution of a reorganization could also be interpreted as a business process. Due to the fact that a reorganization involves many internal and external staff, multiple functions must be executed and countless documents will be created and used, it might be advisable to plan this process as well according to the ARIS concept.

The ARIS procedural model describes how to implement the ARIS concept, i.e, carry out the necessary project organization, create the functions and respective documents (data) and create output packages in ARIS. The ARIS procedural model describes the ARIS concept using ARIS.

Procedural models can also be developed at the type level (modeling level 2), serving as models for a real-world project. Procedural models for BPR, software development, implementing workflow systems or standard software solutions are available in applications and professional publications. These are the models for creating procedures relating to specific projects.

It does not matter if procedural models are not described in detail. For example, Hammer/Champy's BPR procedural model *(see Hammer/Champy, Business Reengineering 1995, pp. 153)* contains only 5 functions: mobilization, diagnosis, redesign, transition and, simultaneous with the first 4 phases, change management. On the other hand, a wide range of procedural models are available for software development, such as the V model of the Coordinating and Counseling Office of the German Government for Information Technology in Federal Administration (*see KBSt, V-Modell 1992*). Various models have been developed for workflow implementation as well *(see Galler, Vom Geschäftsprozeßmodell zum Workflow-Modell 1997)*. To facilitate the implementation of standard software solutions, particularly SAP/R3, SAP and a variety of consultancies offer detailed procedural models *(Keller/Teufel, SAP R/3 prozeßorientiert anwenden 1997, see pp. 177; Meinhardt, Geschäftsprozeß-orientierte Einführung von Standard-Software 1995; Kirchmer, Implementation of Standard Software 1998; Plattner, Products & Organization 1997)*.

In addition to procedural models for miscellaneous modeling goals, models for subordinate modeling objectives, such as data modeling, are commercially available *(see Szidzek, Datenmodellierung-Vorgehensmodell 1993)*. The SOM procedural model, developed by Sinz/Ferstl, focuses on deploying object oriented methods in business process modeling *(see Ferstl/Sinz, SOM 1993, pp. 27)*.

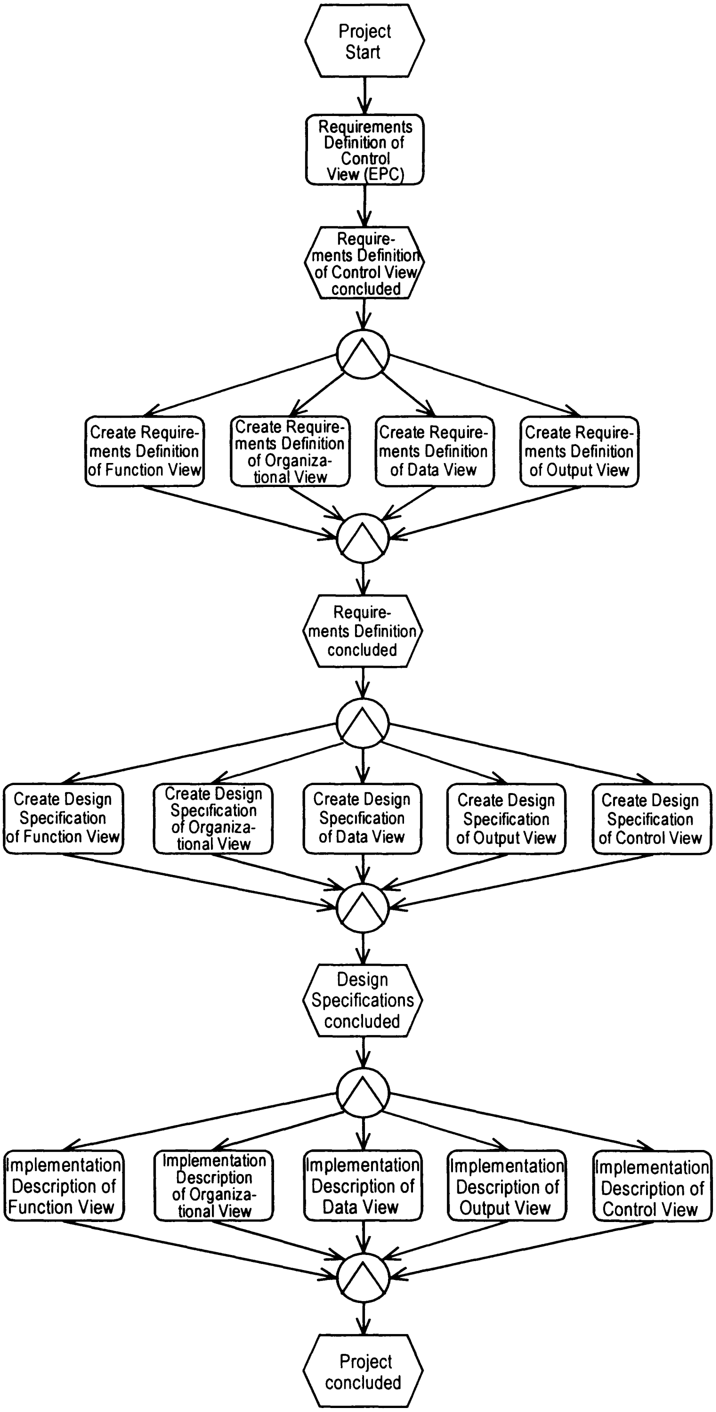


Fig. 21 EPC of an ARIS procedural model draft

At first glance, the ARIS procedural model follows the views and phases of the life cycle, as defined in ARIS. Using these objects and the respective methods, however, the ARIS procedural model can be implemented in much greater detail. Fig. 21 depicts the procedural model as an EPC, consisting of functions and events. The ordering relationships enable the function, organization, data, output and control views, respectively, of a single phase to be executed simultaneously. The requirements definition begins with the control view, i.e., a description of the process. When dividing the processes into small and specific steps, overlapping of detailed areas is possible. Within the processes, various detailing levels can be determined; however, it should be understood that a rigid phased concept, such as in a waterfall model, is not binding. Rather, development forms, such as prototyping, can be illustrated by certain ordering relationships.

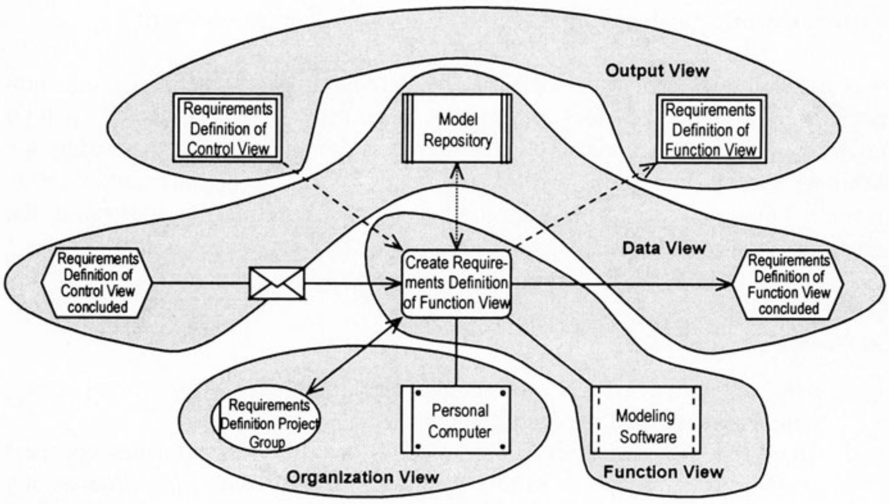


Fig. 22 ARIS views of "create requirements definition function view"

In addition to functions and events, other elements of describing processes, such as the organizational units, data and output involved, can be included. Fig. 22 depicts the "create requirements definition of function view".

- **Data views** contain the milestones of the procedural model, i.e., events and messages initiating or finishing a process, and of environment descriptions, such as models stored in the model repository.
- **Function views provide** the building blocks of the IT architecture. The respective software, whether it be word processing or modeling tools, is allocated to them.
- **Organization views** describe the departments, employees and machine resources necessary for the individual processes.

- **Output views** define the inputs and outputs of function execution, i.e., control and function models.

Up to now, we have only discussed procedural models at the requirements definition level. Due to the fact that they contain IT objects, such as model repositories, personal computers and application software (word processing, modeling software, etc.), procedural models can also be transferred to the design specification and implementation description phases.

Design specifications for repository databases, PCs or modeling software are determined in the design specification. Their IT-specific realization is then described in the implementation description. If the ARIS Toolset is used, the design specification is the phase in which a decision is made, whether single or multiple user versions are to be deployed. The implementation description level is where real-world parameters and actual database products are defined.

Project-specific procedural models can be derived from the general procedural model. For example, if a sales information system is being reengineered, general functions, such as "create data view requirements definition", are enhanced by the addition "create data view and requirements definition of *sales information system*". This compares to the transition from type modeling to illustrating the instances. On the other hand, when abstracting the application relationship in the ARIS procedural model, this results in the general ARIS meta model at modeling level 3.

Let us review:

1. The respective ARIS architecture, describing the business processes, determines the ARIS procedural model.
2. Due to the fact that procedural models can be regarded as business process models, they can be described by the same ARIS concept views as any other business process.
3. The deliverables of every function in the procedural model are stored in the ARIS repository.
4. The repository schema, i.e., the ARIS information model, is the standard for storing results such as the database design for the function, data, organization, output and control models, respectively.
5. The reference model for the procedural model is also captured in the ARIS repository.

Fig. 23 reviews how the ARIS procedural model relates to the ARIS concept, this time using the example of a sales organization. The procedural model is stored in the ARIS repository and defined at level 2. Its logic is determined by the ARIS house and the ARIS information model. As a single project process, the application-oriented procedural model is an instance of the general procedural model for developing business processes, placing this project process at modeling level 1. The procedural model in the repository is modified accordingly. The dashed lines illustrate how the meta level determines subordinate levels: the solid

line shows the deliverables of the procedural model, resulting in the sales models. Describing general sales processes, these models are situated at modeling level 2.

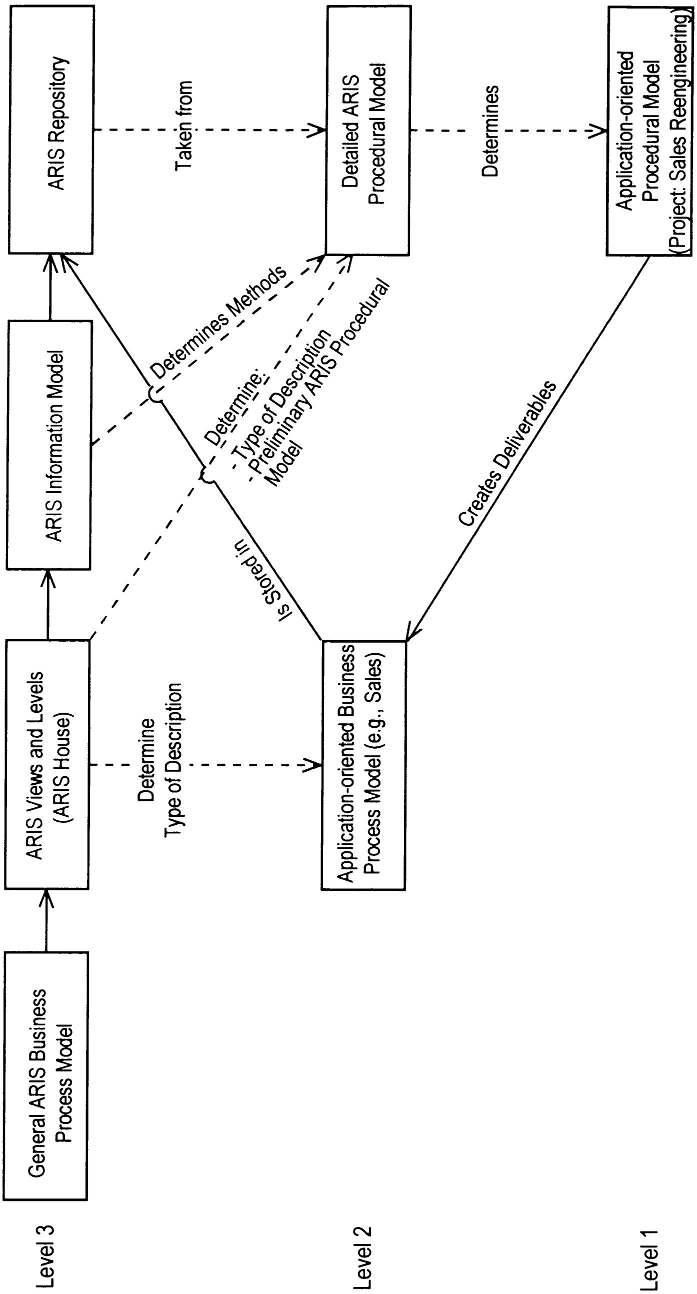


Fig. 23 Relationship between the ARIS concept and the ARIS procedural model

D Business Process Management Using ARIS (ARIS House of Business Engineering)

The ARIS concept paves the way for engineering, planning and controlling business processes. We will now analyze the benefits mentioned in chapter A, plus the business, organizational and IT aspects of its implementation.

ARIS house of business engineering (HOBE) enhances the ARIS process architecture by addressing comprehensive business process management, not only from an organizational, but also from an IT perspective. We will show how ARIS supports business management in the design and development stages, using ARIS-compatible software tools.

Because business process owners need to focus on the „one shot" engineering and description aspects of their business processes, ARIS HOBE provides a framework for managing business processes -- from organizational engineering to real-world IT implementation, including continuous adaptive improvement. HOBE also lets business process owners continuously plan and control current business procedures and devote their attention to continuous process improvement (CPI) (*see Scheer, Workflow-Systeme 1997; Scheer, ARIS - House of Business Engineering 1996; Thome/Hufgard, Continuous System Engineering 1996*). Comprehensive industry expertise in planning and controlling manufacturing processes is a fundamental component of HOBE. Objects such as „work schedule" and „bill of material" provide detailed description procedures for manufacturing processes, while production planning and control systems in HOBE deliver solutions for planning and controlling manufacturing processes. Many of these concepts and procedures can be generalized to provide a general process management system.

- At level 1 (**process engineering**), shown in Fig. 24, business processes are modeled in accordance with a manufacturing work schedule. The ARIS concept provides a framework which covers every business process aspect. Various methods for optimizing, evaluating and ensuring the quality of the processes are also available.
- Level II (**process planning and control**) is where business process owners' current business processes are planned and controlled, with methods for scheduling and capacity, and (activity based) cost analysis also available. Process monitoring lets process managers keep an eye on the states of the various processes.

- At level IV (**workflow control**), objects to be processed, such as customer orders with appropriate documents or insurance claims, are delivered from one workplace to the next. Electronically stored documents are delivered by workflow systems.
- At level IV (**application system**), documents delivered to the workplaces are specifically processed, i.e., functions of the business process are executed using computer-aided application systems -- ranging from simple word processing systems to complex standard software solution modules--, business objects and java applets.

HOBE's four levels are linked with one another by feedback loops. Process control delivers information on the efficiency of current processes. This is where continuous adaptation and improvement of business processes in accordance with CPI begins.

Workflow control reports actual data on the processes to be executed (amounts, times, organizational allocations) to the process control level. Application supporting modules are then started by the workflow system.

In the fifth component of the HOBE concept, levels 1 through IV are consolidated into a **framework**. Frameworks contain information on the respective architecture and applications, configuring real-world applications from the tools at levels II and III as well as application information from the reference models (level I). Frameworks contain information on the composition of the components and their relationships *(see Pree, Komponentenbasierte Softwareentwicklung 1997, p. 7).*

Software at process engineering and process planning levels supports the business-organizational view of the business process owner, whereas workflow control and application system levels refer to the IT-specific implementation. With software installed at all four levels, ARIS life cycle model is applicable to every one of them. Therefore, at every level it is possible to specify any software system in the requirements definition, design specification and implementation description, respectively. Relationships between the levels outlined by the HOBE concept are discussed primarily at the requirements definition level, for example, how to logically pass a process model from level II to a workflow model at level III. This would also be where compatibility of the modeling tool design specification at level I with the workflow system at level III, even implementation aspects, would be discussed. The levels of the HOBE concept are perpendicular with the phases of the ARIS life cycle.

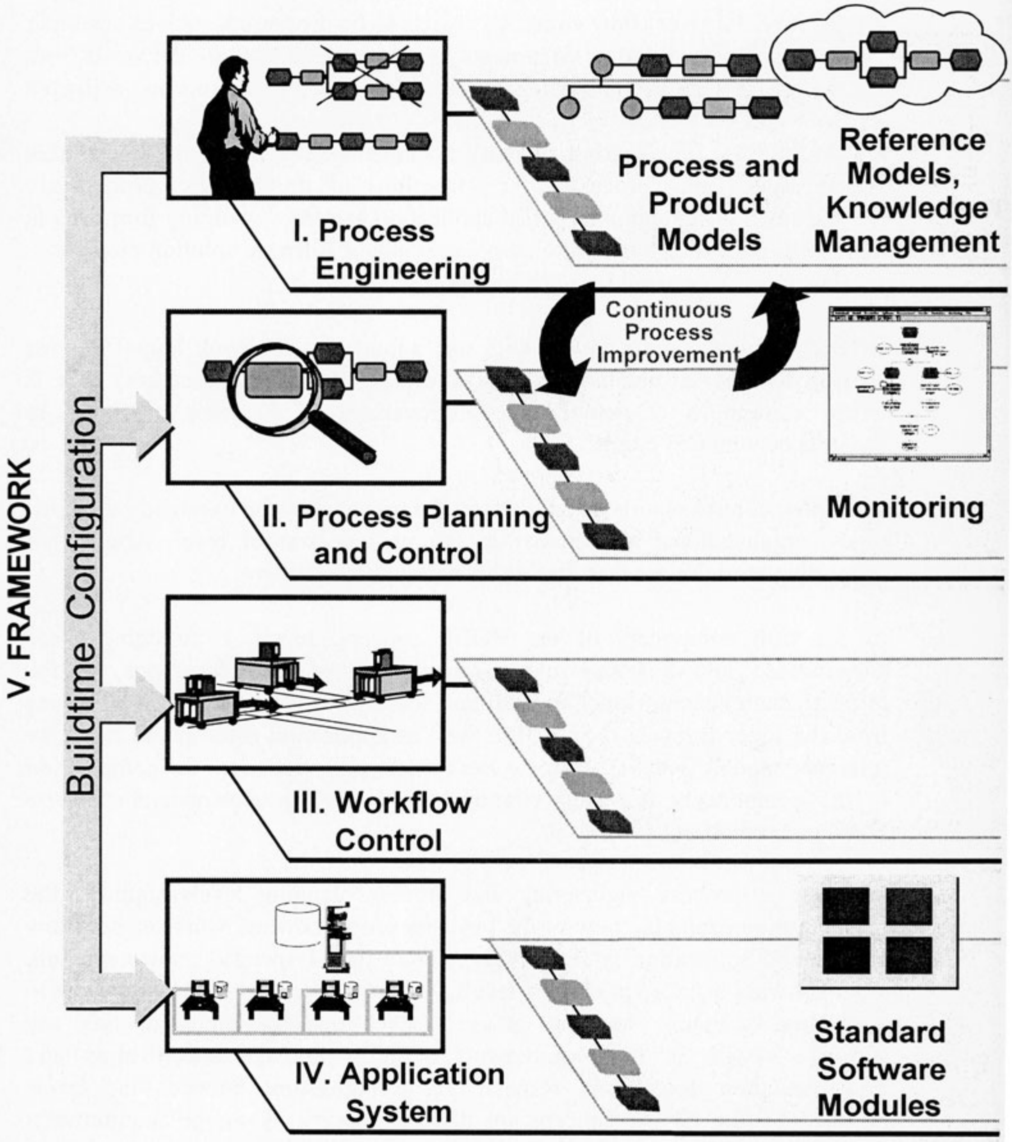
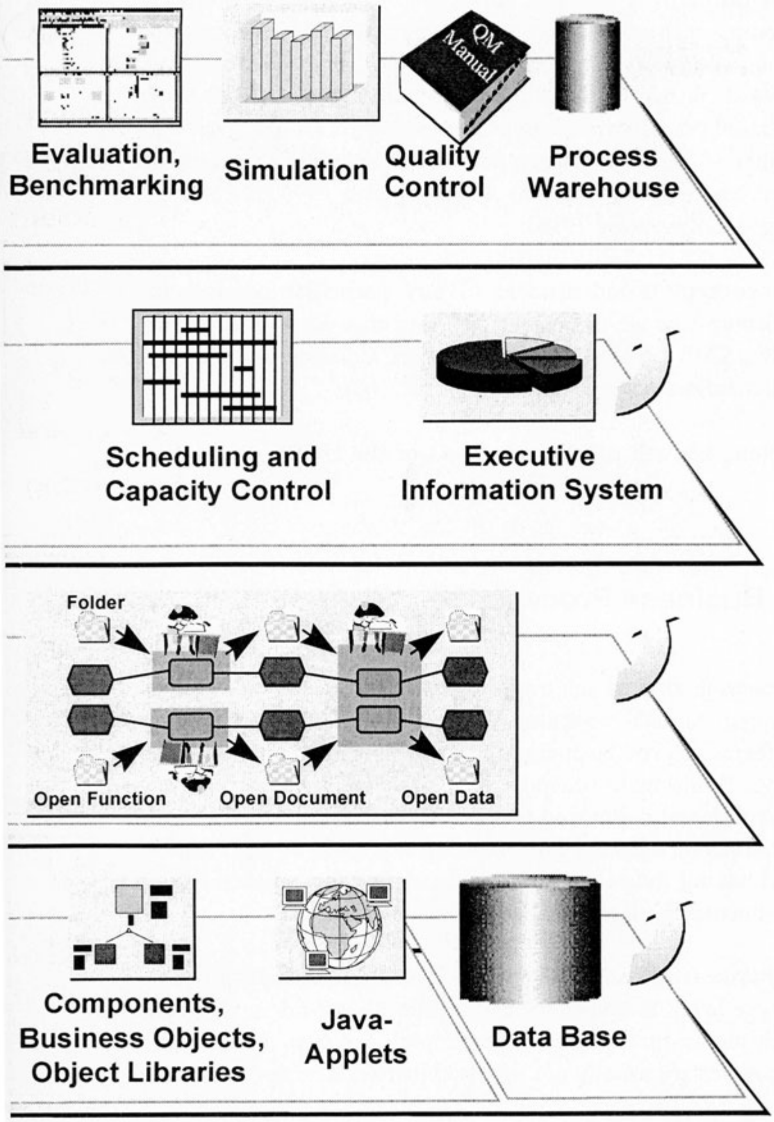


Fig. 24 Process management with the ARIS - house of business engineering concept


QM Manual
Evaluation, Benchmarking
Simulation
Quality Control
Process Warehouse
Scheduling and Capacity Control
Executive Information System
Folder
Open Function
Open Document
Open Data
Components, Business Objects, Object Libraries
Java-Applets
Data Base

The HOBE concept is primarily a concept, although it can also be used as a framework for developing real-world software products. The HOBE concept was premiered by the author at the Saarbrücken Workshop (Saarbrücker Arbeitstagung) in 1994. It is the standard corporate architecture at IDS Prof. Scheer GmbH and is based on real-world application expertise. It has also been suggested to many other software providers, has been presented in scores of talks held by the author, and was the topic of a white paper in 1996 *(see Scheer, ARIS - House of Business Engineering 1996)*.

Although the HOBE concept is independent of any particular commercial products, for practical purposes we are presenting it in this work, along with various ARIS products, SAP R/3 and Siemens-Nixdorf ComUnity . We will also be referring to other software concepts.

In the following chapters, we will discuss the levels of the HOBE concept in greater detail.

D.I Engineering Business Processes

Business process engineering aims to achieve the greatest efficiency possible in terms of business-organizational solutions. Organizational departments, reengineering project teams or even business process owners can be responsible for process engineering. While work schedule development for manufacturing processes might be institutionally allocated to a certain department for years as job preparation, other kinds of business processes are not quite as regimented. We would recommend having the same entities responsible for engineering as are responsible for the business processes themselves.

Generally, enterprise business processes, such as a typical purchasing process, are engineered at the type level. Subtypes for certain subforms (orders for spare parts, normal parts or just-in-time parts, for example) can also be created. However, ordering processes are usually not modeled just because specific parts need to be ordered.

On the other hand, work schedules for specific parts in manufacturing processes are indeed documented. This is due to the fact that process descriptions are not only used to support fundamental organizational rules, but also for direct process execution. The more process documentation is utilized for executing business processes, such as for workflow control, the more descriptions for process instances become necessary.

When engineering optimal business processes, reference models can be included, along with available knowledge on best practices. It is also possible

to compare alternative procedures (benchmarking) or carry out simulation studies or quality evaluations. We will now discuss engineering aids.

D.I.1 Modeling Product and Business Processes

Business process engineering starts with strategic enterprise planning, where product groups and core corporate processes are determined. Products of course being created by processes, product areas determine the necessary business processes.

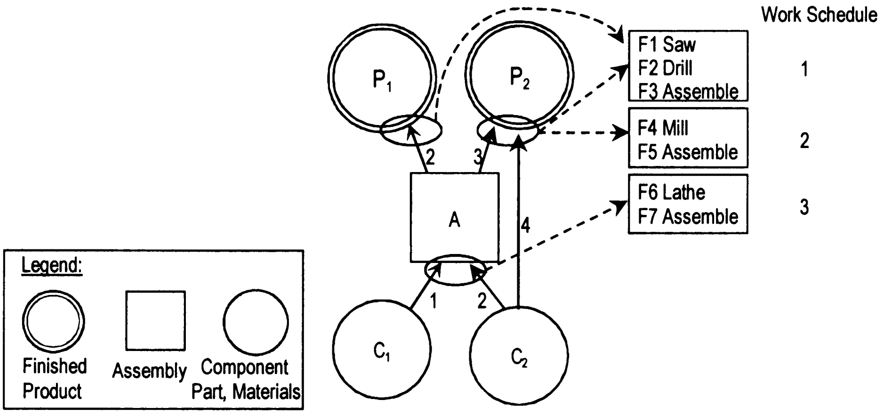


Fig. 25a Product and process model

The terms „bill of material" (BOM) and „work schedule" do a good job of describing the interrelationships between product and process models in industrial manufacturing. Fig. 25a offers an example. The bill of material describes the composition of finished products (P1, P2 in the example), consisting of assemblies (A) and component parts (C1, C2). In the work schedule, manufacturing processes are allocated to every part to be manufactured. A work schedule comprises the operations (functions) to be executed. Work schedules are generally shown in tables. Several alternative work schedules can be defined for one component part such as P2 for work schedules 1 and 2.

Independent product and process descriptions make it possible to allocate one (standard) work schedule to multiple parts (in the example, work schedule 1 is allocated to finished products C1 and C2). Finished products may be different as far as subordinate parts are concerned, as long as this does impact the manufacturing process. For example, certain products in the chemical industry are made up of identical components, but with different colors, resulting in different products with identical manufacturing procedures. Independent

definitions of products, process models and their unencumbered allocation are key to non-redundant data management.

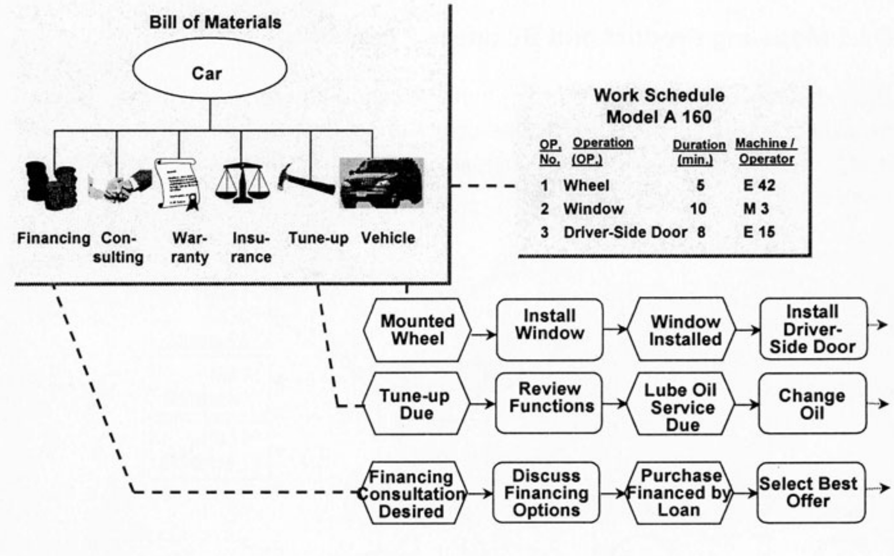


Fig. 25b Equivalent product and process descriptions for different services

In manufacturing companies, detailed product and process descriptions are used only for physical products. However, there is a growing trend to bundle physical products with services, as illustrated in Fig. 25b, with services such as car insurance or car financing. A work schedule table is allocated to the physical product with EPC processes defined for creating services. These processes describe the required function sequence in order to close the appropriate insurance policy or credit application.

Complete product and process modeling of every physical and non-physical product makes unified business process management possible. Comprehensive calculation procedures are a prerequisite for determining product costs and for achieving synchronized planning and control of business processes. We will later elaborate on this concept. Not only are processes key in the manufacturing of products, but process forms also determine product types. Fig. 25c, top row, shows the various processes for the product „fine restaurant dining", consisting of the functions order, serve, eat and pay. Meals in fast-food restaurants, however, consist of the following functions: order, pay, (self-) service and eat *(see Pentland, Process Grammars 1994).*

Here, the process type, especially the sequence of the functions, greatly impacts the type of product. Process innovation thus also leads to *product* innovation.

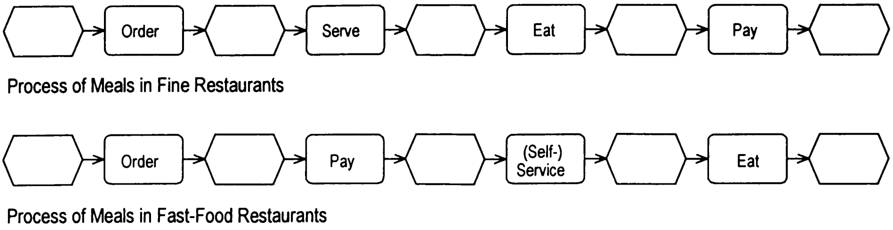


Fig. 25c Relationship among product and process innovation

D.I.2 Reference Models

Reference models, which can be developed in real-world situations (best practices) or theoretically, document process know-how that can be utilized for modeling. We can distinguish between procedural models or the implementation of standard software, and business models such as for order processing or product introductions.

Models can be specialized for vertical markets (resulting in vertical market reference models). ARIS concept reference models, developed by consultancies with expertise gained in customer projects, are available for practically every vertical market. Thus, documented process expertise results in the development of commercial products.

Reference models can be quite comprehensive, consisting of hundreds or thousands of model objects. This is why various levels of aggregation are used.

Reference models provide enterprises with an initial process engineering solution, letting them determine the degree of detail of the model and the business content. Adapted to company-specific requirements, reference models evolve into company-specific models. Actual case studies have shown that the use of reference models in organizational projects can reduce time factors and costs by more than 30%.

Reference models provided by software vendors as software documentation (the most comprehensive model being SAP's R/3 reference model) benefit the customer by utilizing business process know-how, providing the opportunity to compare business software solutions or pinpointing positive or negative implementation issues.

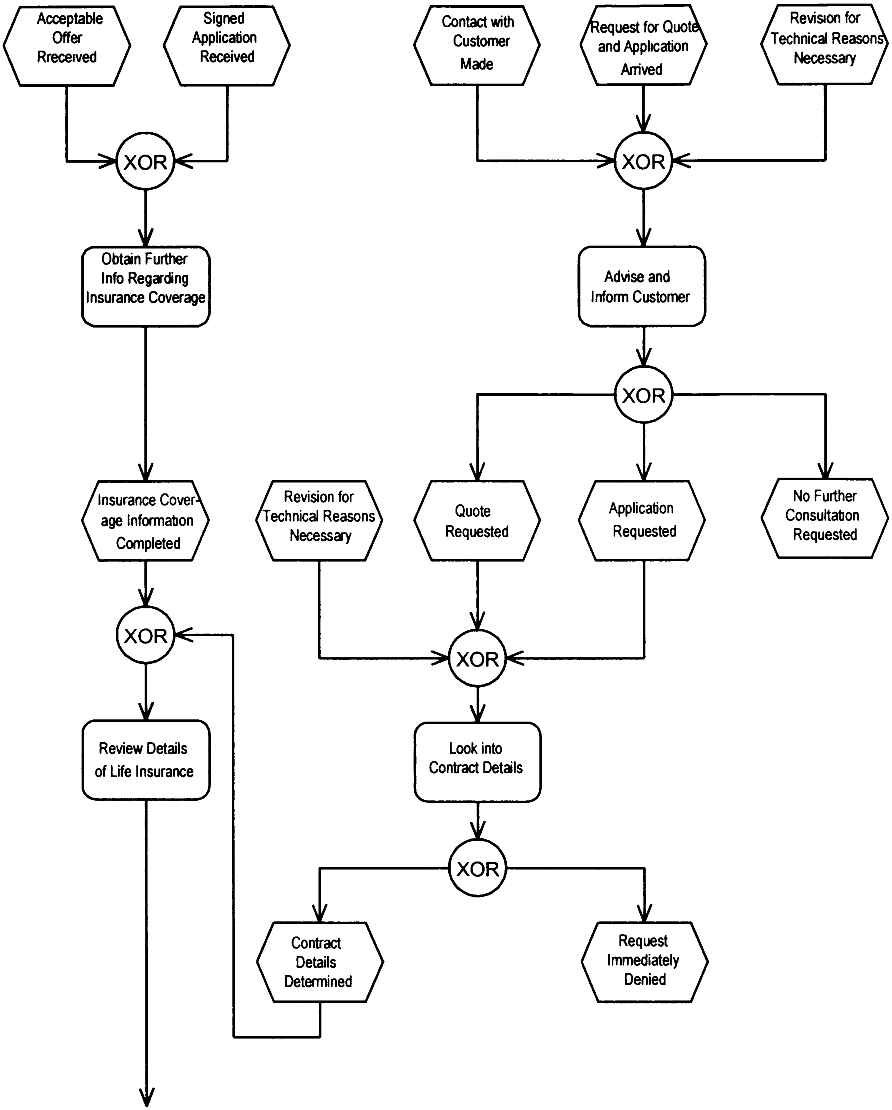


Fig. 26a Excerpt of an EPC from an ARIS insurance reference model designed by KPMG

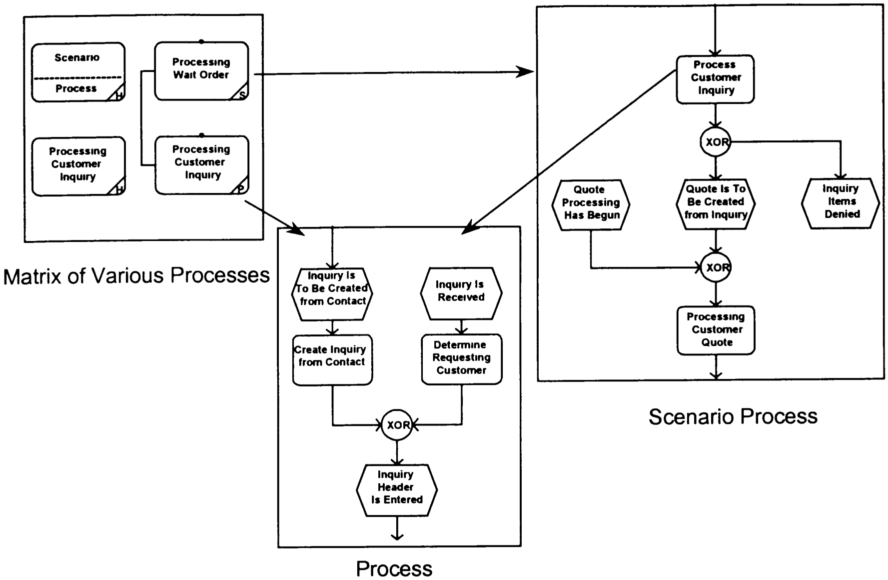


Fig. 26b Excerpt of an EPC from an R/3 reference model

Fig. 26a-b show excerpts of a vertical market reference model for insurance companies, developed by KPMG Consulting, and an SAP R/3 reference model. Both were developed largely in accordance with the ARIS concept. The insurance reference model consists of 543 functions, the SAP R/3 model is several times larger. Both reference models include function, data and organization models, in addition to process models.

IDS Prof. Scheer GmbH offers additional reference models designed in accordance with the ARIS concept for the manufacturing, energy, paper processing, financial and chemical industries and for public services as well.

D.I.3 Knowledge Management

Process know-how is increasingly being regarded as an important component of overriding corporate knowledge management. Corporate knowledge includes know-how regarding the products, technologies, organizational procedures and rules as well as the individual know-how of each individual employee.

Documenting, storing, utilizing and enhancing this basic know-how is a key task of knowledge management.

Storing a company's process know-how in a process warehouse is an important step towards managing knowledge management. In its organizational view, the

ARIS concept provides the functionality for capturing corporate structural information on technical procedures and resources and even on the know-how of individual employees.

The data view captures knowledge documents stored in traditional media as well as text, voice, image and video documents.

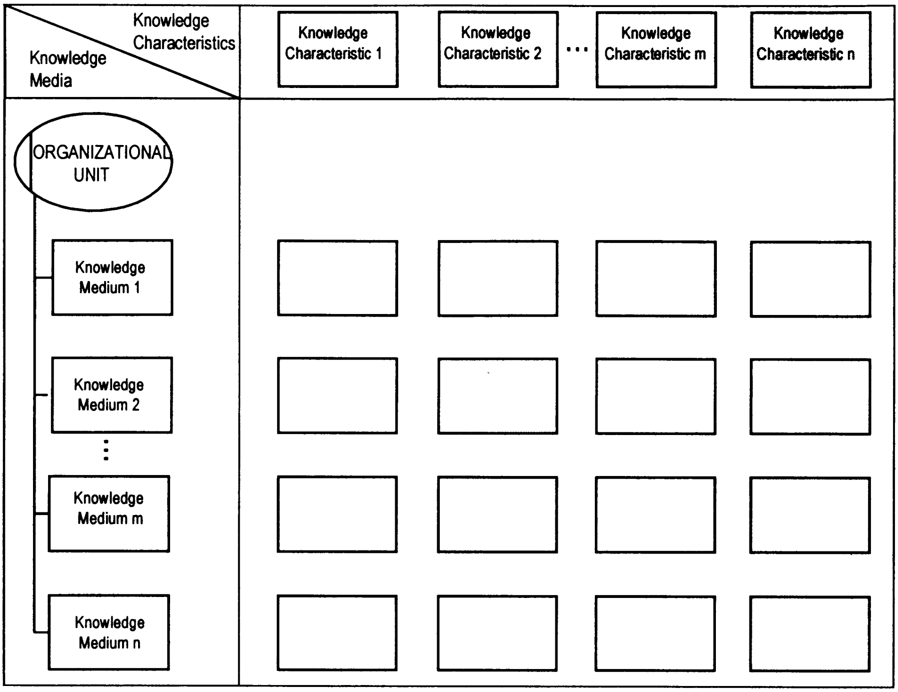


Fig. 27 Knowledge topography
(from Scheer/Bold/Hagemeyer/Kraemer, Informationssysteme im Wandel 1997, p. 27)

The control view is where links between different knowledge types are displayed. For example, the profile of an employee's know-how or the expertise an employee obtains while carrying out a certain function can be linked to other functions. The various hierarchies of knowledge workers in a company (individuals, groups, the company itself, a group of companies, company alliances) can be defined in the ARIS organizational view and then linked to knowledge types in other views (see Fig. 27). Thus, ARIS becomes the framework for „organizational memory" or for a „knowledge warehouse".

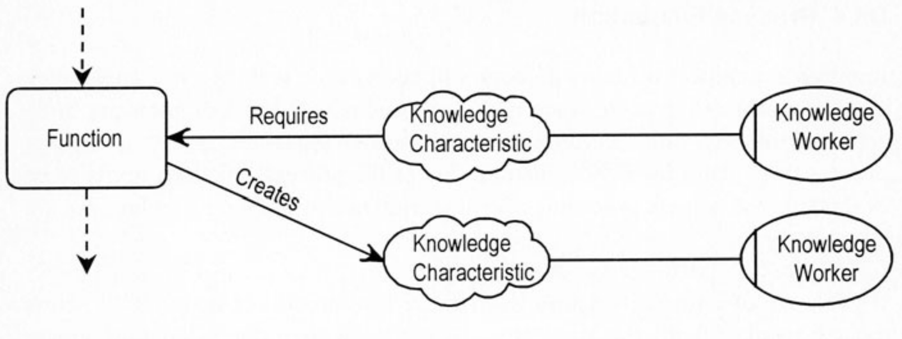


Fig. 28a Knowledge management in an EPC

Modeling methods should be enhanced by the respective knowledge types or knowledge worker objects used or created in a function (see Fig. 28a). By linking these objects in ARIS, access paths turn into „knowledge maps". Knowledge warehouses detect and eliminate knowledge deficits, unused knowledge, lack of knowledge transparency, inefficient knowledge distribution or uncoordinated accumulation of knowledge *(see Probst/Raub/Rombardt, Wissen managen 1997; Myers, Knowledge Management and Organizational Design 1996)*. Fig. 28b is an instructive example of a knowledge life cycle where individual phases can be evaluated by positive or negative factors.

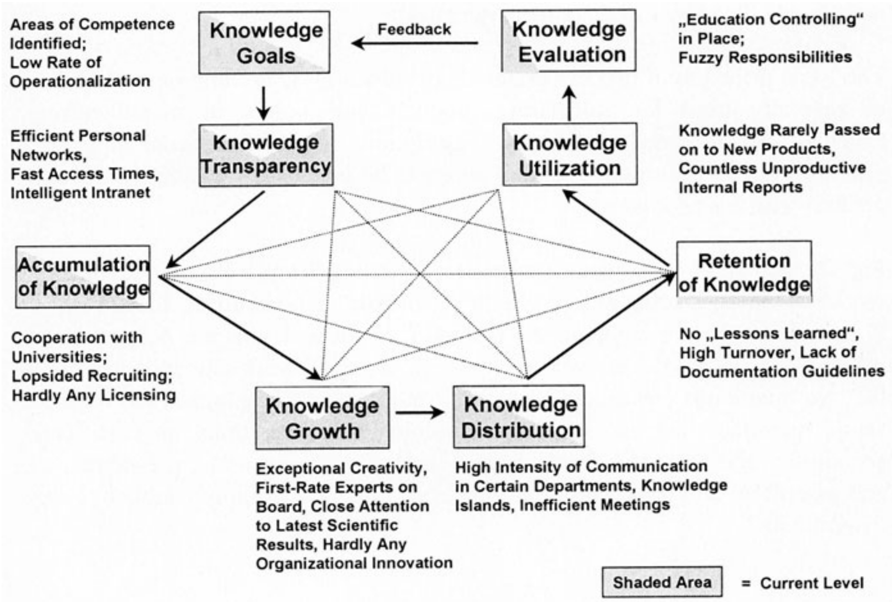


Fig. 28b Knowledge profiles in a company
(from Probst/Raub/Rombardt, Wissen managen 1997, p. 346)

D.I.4 Process Evaluation

In order to engineer business processes in accordance with specific goals, they must be evaluated in accordance with process goals. When key goals are to be achieved in, say, the electronics or automotive industries, (such as „lower process throughput by 50%"), the duration of the process functions needs to be evaluated. Assessment procedures derived from network diagram techniques are appropriate.

If goals are of a financial nature, such as „reduce process costs by 30%", costs must be linked with the processes. However, with a focus on cost center accounting, current business cost accounting systems are more geared towards functional points of view. For example, target cost accounting aims at optimal control of cost centers created in accordance with functions. On the other hand, costs of the business processes are not known, which is why activity based costing opens new doors. The business role of activity based costing has been a topic of professional discussion for some time *(see Glaser, Prozeßkostenrechnung 1992; Horváth/Mayer, Prozeßkostenrechnung 1989; Kloock, Prozeßkostenrechnung 1992, among others)*, although we do not wish to elaborate on it in this book.

Activity based costing is based on the basic principle of segmenting business processes into elementary subprocesses. The average costs of executing each subprocesses at one time are determined. The cost of the entire business process is then calculated by calculating charge rates.

The basic principle of process oriented cost charging is nothing new. Rather, it is generally used for calculating products and orders in manufacturing. Calculations are based on process descriptions (bills of material and work schedules). The information gained here can be adapted to evaluating costs of general business processes.

Fig. 29 depicts a manufacturing process resulting from bills of material and work schedules, leading to step one in calculating manufacturing costs. Product P is made up of two components, C1 and C2. These details are defined in the bill of materials for P. For every component, there is a work schedule describing the two operations (setup and assembly / mounting). The planned duration for setup, manufacturing and assembly is allocated to operations. In cost center accounting, the cost rates for each cost center (in this example, pre-fabrication and assembly) and for specific references (setup and assembly / mounting) are determined.

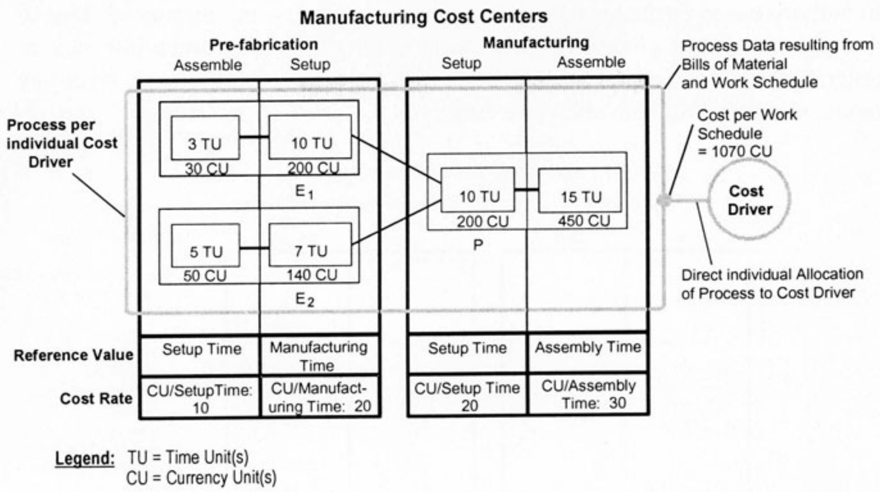


Fig. 29 Calculating a manufacturing process

After calculating manufacturing costs, they are multiplied by the duration of the usage of their components. Usage per operation is determined individually by details in the work schedule. For example, component C1 needs 3 CU for setup and 10 TU for assembly. Multiplied by the cost rate of 10 CU/TU or 20 CU/TU, this equates to 30 or 200 CU, respectively, for the execution of the functions. By adding up all the function costs, the manufacturing of one unit of P1 costs 1070 CU. Since P1 is the cost driver in question, the costs of the manufacturing process are equal to the manufacturing costs of the P1 cost driver.

Manufacturing features work schedules and bills of material with detailed descriptions of the processes. Similar descriptions are generally not (yet) available for administration. This is why it is not possible to directly translate charge rate calculation to processes in these areas.

In traditional cost accounting and for processes in indirect output areas, it is also possible to create differentiated reference values, representing the functions executed within them. For example, in purchasing, there is a reference value „number of orders“, in administration, „number of reviewed invoices“ and in sales, „number of consignment orders“. These reference values can be used for cost center oriented planning, although the interrelationship with the individual cost driver is not known in the calculation, and therefore only flat rates of these activities' usage can be deduced. This is why aggregated overhead rates (say, based on manufacturing costs) are calculated, although this is criticized as being too general. Activity based costing offers new opportunities to resolve these issues.

In activity based costing, process descriptions must first be introduced. Due to the high degree of complexity, the wide variety and sometimes low rate of reiteration in office procedures, however, processes at the type level are modeled rather than individual processes.

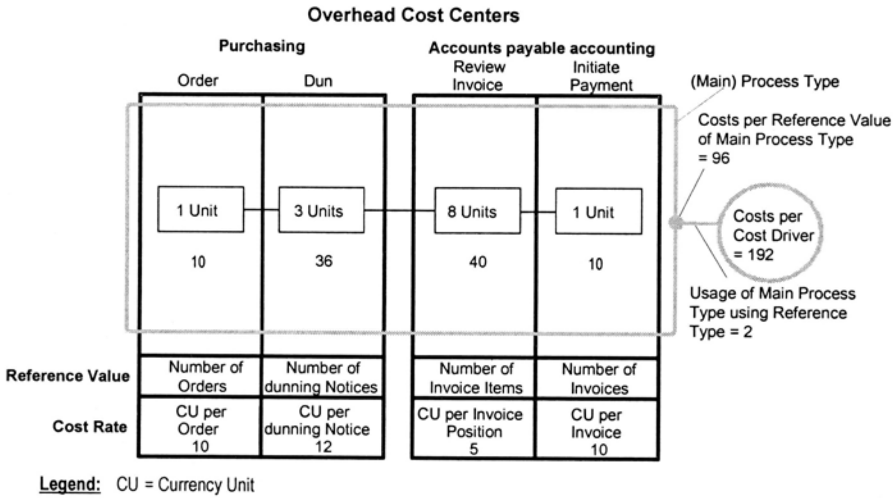


Fig. 30 Information base of activity based costing in office processes

Typical subprocess types, consolidated across cost centers into main process types, are defined for each cost center (see Fig. 30). The designation „subprocess" indicates that this is a component of a business process. If it is not broken down further, it equates to the designation „function". In the example in Fig. 30, the main process „purchasing" consists of four subprocesses: order, dun, review invoices and initiate payments.

As in manufacturing, subprocesses correspond with the designation „operation". Cost rates (i.e., process cost rates such as costs per typical purchase order, costs per typical dunning statement, etc.) can be determined for each subprocess. For each subprocess, measures of usage are allocated to the main process. These details are shown in the function boxes in Fig. 30. Multiplying the process cost rates by the usage measures results in process costs per subprocess. When these figures are added up, this results in costs per main process -- in this case, per typical purchase order. If these process costs are to be used for the calculation, the cost drivers must be charged. This requires defining a reference value which indicates how many main processes of the type „purchasing" are linked with one unit of the cost driver. In our example, usage equals 2, resulting in 192 CU per cost driver unit.

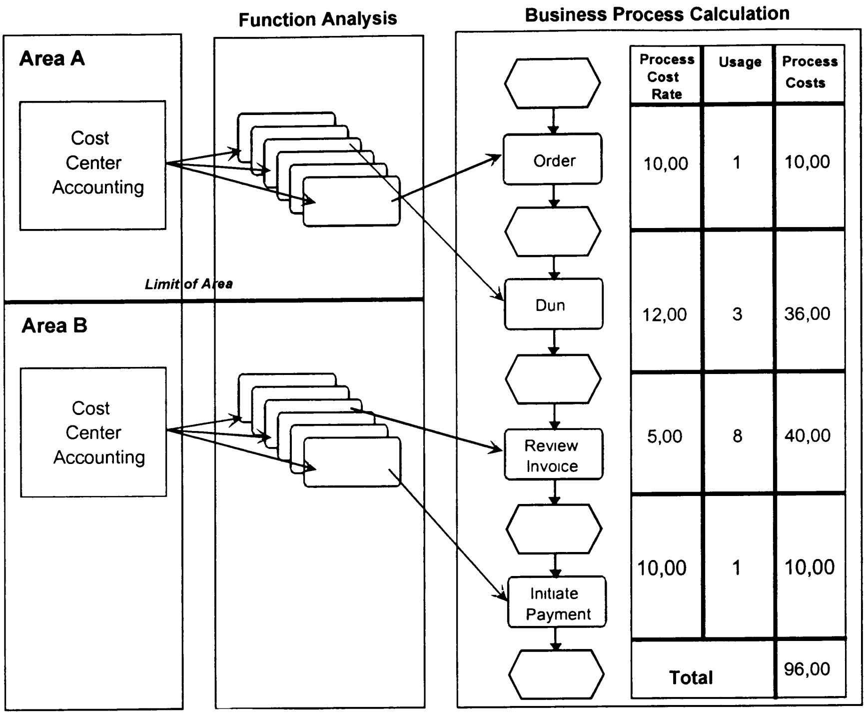


Fig. 31 Calculating business processes

Based on process modeling, the basic principle of charge rate calculation can be expanded to include indirectly productive areas of output.

Allocating process costs in accordance with usage can only be as accurate as the reference value of the main process linked to the cost driver and the value of usage determined. It is not possible to directly allocate a main process unit to a cost driver unit, as opposed to the procedure in manufacturing costs.

Process oriented allocation of overhead costs therefore prevents cryptic flat rates for these values.

Existing operational and cost center oriented cost accounting systems are not replaced by activity based costing, but rather are integrated in the process calculation as information sources and recipients. Fig. 31 shows a scheme of an integrated model based business process calculation, linked with traditional cost center accounting.

This is based on a traditional cost accounting system with detailed cost center accounting. First, functions in cost centers are analyzed and their reference values determined. Then, the process cost rate per function is calculated.

The function structure, depicted in Fig. 31 as an EPC, is adapted from business process modeling. Usage measures per process chain function are multiplied by the process cost rates. Costs of the (main) process can be deduced from the total function costs. Process and figures in Fig. 31 correspond with the example in Fig. 30.

D.I.5 Process Benchmarking

Benchmarks can provide reference points for engineering highly efficient business processes. Target or orientation values are obtained by comparing a company's business process to a similar process, also called „benchmarking". When stacking up business processes against each other, comparative processes of one's own company can be used, for example by comparing business processes of various departments or sales processes of sales offices. Processes of competitors in the same industry can also be used, for example generally available reference processes compiled by consultancies, or similar processes in different industries. For example, a pit stop process of an Indianapolis 500 racing team could provide valuable input for the aircraft turnaround process of a particular airline.

In a nutshell, there are multiple sources providing process know-how, ranging from scientific publications to conference reports, commercial benchmark data, studies of associations and services of consultancies.

Pointers on how to engineer a company's processes can be gleaned from the deviation between the process values of the benchmark partner and the company's own measures. Benchmarking target values can be financial, time-related or volume-related indicators such as process costs, throughput times or input/output values, although more qualitative statements regarding customer satisfaction are important, too.

Although benchmarking is not a new concept in business, it does offer new ideas on how to streamline and speed up business processes.

Furthermore, various publications *(see Küting, Benchmarking von Geschäftsprozessen 1996; Aichele, Kennzahlenbasierte Geschäftsprozeßanalyse 1997)* show to what extent evaluation criteria for business process engineering have been studied. Fig. 32 provides an overview of benchmarking criteria.

QUANTITATIVE AND QUALITATIVE BENCHMARKING-CRITERIA

Productivity Mass
Product output per number of employees; Product output per resource usage; Costs per "good" product unit; Number of completed orders per work hour; Added value per employee; Cost percentage of value added activities...

Quality Mass
Percentage of scrap; Percentage of postprocessing (products, working hours); Percentage of shipments with defects (products); Percentage of returns; Warranty costs; Availability and accuracy of information...

Time Requirement Mass
Percentage of punctual deliveries; Lead time for product construction; Lead time for delivery; Number of delayed deliveries; Order range; Time required for changeover; Time required for review; Percentage of non-productive time...

Customer Satisfaction
Customers wanting to make renewed purchase; Satisfaction indices; Output to be realistically expected; Purchase recommendations; Perceived functionality; User friendliness...

Paperwork
Time required for order processing; Number of obstacles for customer; Average number of contacts per order fulfillment; Number of defects and postprocessing requirements; Number of permissible exceptions; Report delays in days (products, sales)...

Fig. 32 Selected quantitative and qualitative benchmarking criteria *(from Küting, Benchmarking von Geschäftsprozessen 1996, p. 135)*

D.I.6 Simulation

While it is essential to evaluate activity based costing and benchmarking results for a single business process, multiple alternatives are generated, studied and analyzed in simulation studies in order to engineer the best possible business process.

No methodical enhancements of the business process model are necessary for defining and analyzing the various engineering alternatives in what-if-situations. After analysis, the existing process model serves as the foundation for the simulation.

In dynamic simulations, on the other hand, the dynamic behavior of process alternatives is studied. Individual processes are generated in accordance with the process model, their processing is tracked. Thus, processes are defined at the instance level and their interrelationships are analyzed. This pinpoints any potential delays before any processing begins.

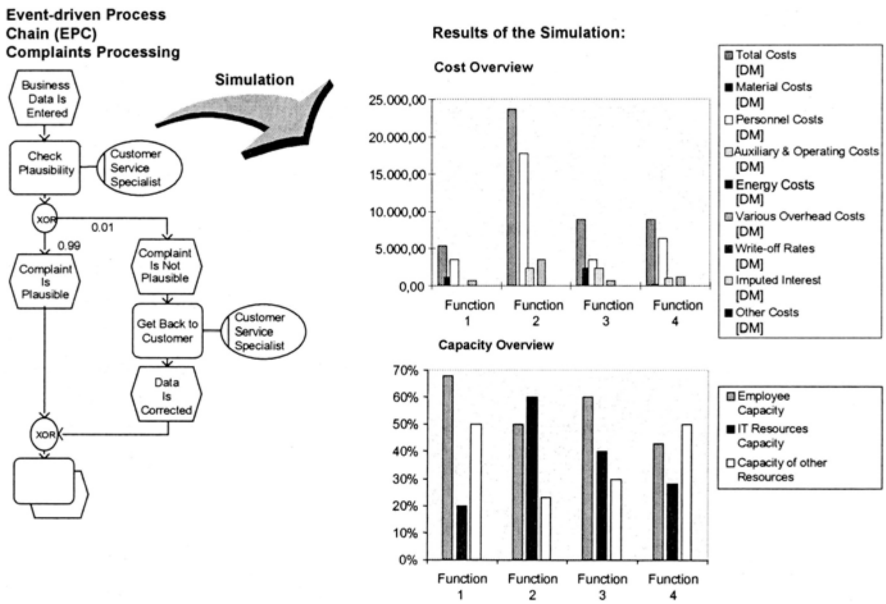


Fig. 33 Example of a simulation

As far as the process alternatives to be analyzed are concerned, it is possible to define various process structures, processes with different function times and operating behavior, respectively, of the respective organizational units. Alternatives are generated individually, in accordance with empirical studies, or randomly and automatically.

The structure of a simulation model can be derived directly from the general structure process, as shown in level I of the HOBE model, and then evaluated by a simulation generator. Fig. 33 depicts an example for evaluating a business process simulation using the SIMPLE++ system, in conjunction with ARIS Toolset. Simulation studies to determine the best way of engineering business processes have been used for a long time in manufacturing procedure studies, for example for determining effective priority rules for heuristic process control *(see Glaser/Geiger/Rohde, PPS 1992, pp. 225)* or for supporting the layout plans of industrial companies. For office processes, however, comprehensive dynamic simulation studies are less common, although they are becoming increasingly popular due to the optimization of administration and service processes. For example, simulation applications are now being used in banks and insurance companies.

D.I.7 Quality Assurance

ISO 9000 definitions include criteria for defining the quality of business processes. Companies can have their adherence to these standards certified. The main idea of these certifications is that the quality of the processes is an indication of the quality of the processes themselves.

All around the world, standards such as ISO 9000 and 9xxx, as well as the more rigid QS-9000 in the automotive industry, are now well established. In addition to certifying adherence to basic standards like ISO 9001, they stress management aspects and pave the way for total quality management (TQM). Key TQM models like the Malcolm Baldrige Award or the European Quality Award (EQA) view business processes as the focal point of their evaluation criteria, enhancing corporate success by focusing on customer orientation *(see Seghezzi/Hansen, Qualitätsstrategien 1993; Seghezzi/Dahlem, Schritt für Schritt zu TQM 1997).*

Efforts towards enhancing quality do not grind to a halt, however, once adherence to ISO 9000 standards has been certified. In order to optimize enterprise processes in accordance with certain goals, TQM requires people to think and act in a process oriented manner and to constantly review and improve existing procedures.

Describing entire business processes with the ARIS concept automatically leads to consistent modeling. Every basic QM element in the ISO 9000 standard can be documented in the ARIS concept. This includes describing organizational responsibilities, identifying and tracing products, purchasing and manufacturing products, steering documents as well as handling, storing, packaging and shipping products. The QM documentation guide, originally designed for external usage, refers to more detailed procedure and operating instructions. These are described as an EPC in a process hierarchy consisting of several levels (see Fig. 34). Cross references within the models are always consistent, even without manual maintenance. This system supports the allocation of the models to 20 ISO 9001 elements, making it possible to automatically create the QM guide, procedural and operating instructions, and the job opening descriptions *(see König/Packowski/Wyler, BPR als Chance 1995).* As a consequence, paper versions of the QM documentation are not necessary.

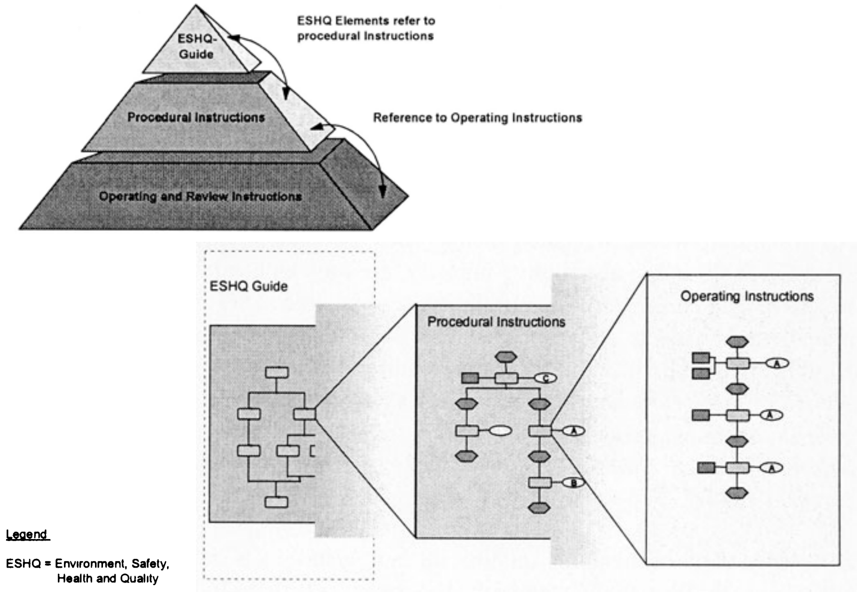


Fig. 34 Levels of QM documentation

Using a modeling tool and storing processes in a process repository ensures meeting the standard requirement that processes should at all times be at the disposal of all the respective persons in the enterprise. Granting user and access privileges guarantees that all the respective users have read-only access to the relevant process. Current data can then be made available to these persons in corporate intranets.

D.I.8 Process Warehouse

The result of systematically capturing, storing and maintaining business process know-how in a repository is called a process warehouse.

Process warehouses are fed from a wide range of project sources in which business processes are analyzed. These projects can include reengineering tasks, ISO 9000 certification, implementation of standard software, activity based costing, etc. When various methods and tools are used in these projects, the content of the models in the process warehouse needs to be consolidated and then merged with other models. In consistent and transparent organizational guides, this process know-how can then be made available to additional projects. Finally, Internet and intranet technology enables distribution in global enterprises.

Because process know-how is a specialty of employees working in operations, this data should be captured and maintained in a decentralized repository. Specific method skills are not required because the focus is on capturing content.

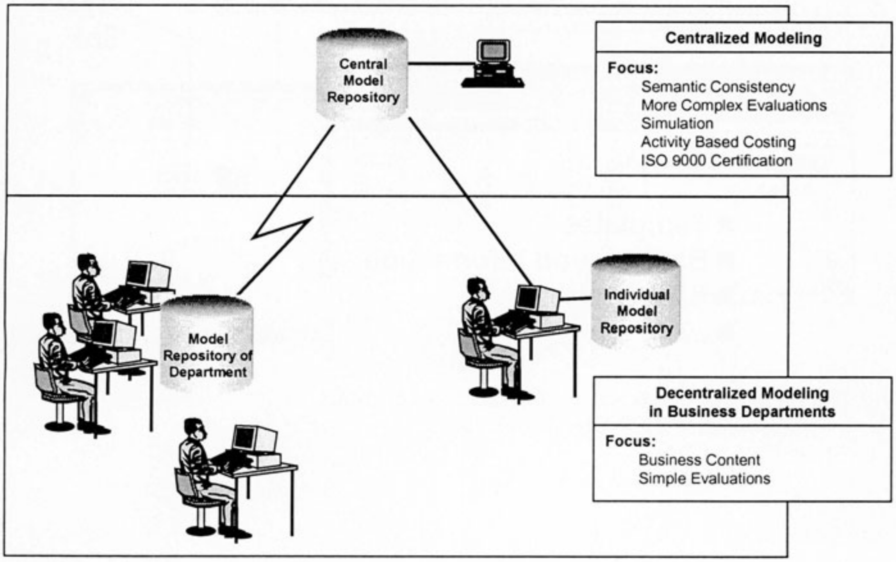


Fig. 35a Centralized and decentralized modeling in a client/server environment *(Source: IDS, ARIS-Easy Design 1997)*

Once captured, this data is consolidated at an overriding centralized level and then methodically analyzed. It is also possible to carry out more complex evaluations such as simulations, activity based costing, etc. Fig. 35a depicts a client/server architecture of this process warehouse concept. Thus, a purely graphical process illustration of an EPC contains only a fraction of the knowledge pertaining to a certain business process. It does not include organizational-, cost-and time-related data that is captured by other means such as in tables. Fig. 35b shows a graphical EPC illustration of a business process, enhanced by multimedia elements such as video, graphics, images and tables.

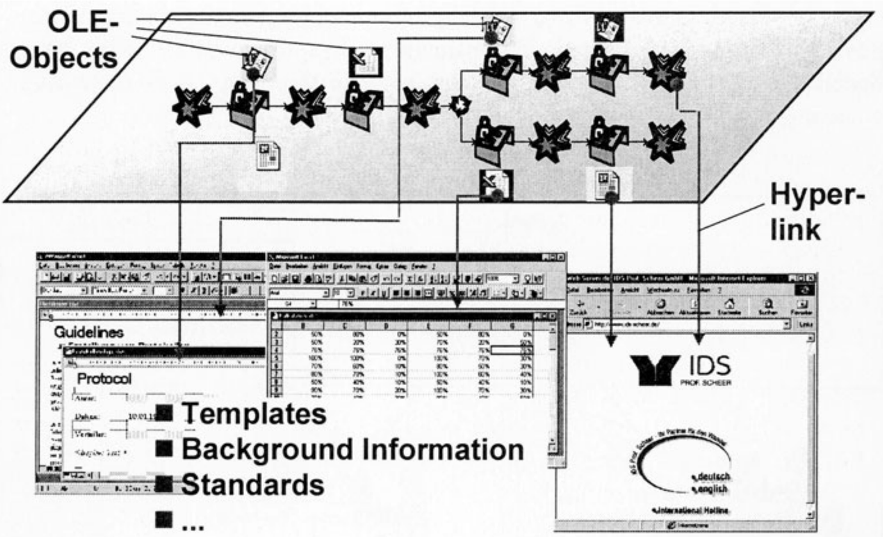


Fig. 35b Illustration of a process with multimedia elements
(Source: IDS, ARIS-Easy Design 1997)

D.II Planning and Controlling Business Processes

Engineering a business process concludes in a kind of template for individual business processes (process instances). In order to be able to plan and control current business processes, the appropriate information must be made available to the persons responsible for the process. In production environments, it is common to plan and control processes. For decades, IT systems have been used for production planning and control. They operate according to a progressive planning concept. First, long-term material and capacity requirements of the forecasted customer orders are determined. Then, short-term manufacturing orders are scheduled, after which the sequence of individual operations is optimized *(for details, see Scheer, Business Process Engineering 1994).*

Planning and control systems are not yet customary in office environments. However, these basic concepts may be applied without having to create a rigid work environment in the office. Output forecasts may be used for office procedural planning, including long-term allocation of personnel, office space and resources. The identification of the necessary process types leads to the number of business processes to be executed.

„Controlling“ processes in an office environment mean that the person in charge can provide customers with information on the status of their individual processes. Resources must be utilized in an optimal manner, process throughput times and process qualities must be constantly monitored. By changing process priorities, resource allocations and processing sequences and in order to achieve the process goals, process owners can manipulate the process. Process monitoring, scheduling and capacity control, and executive information systems (EIS) can provide them with the necessary tools.

D.II.1 Process Monitoring

Process monitoring provides the employees involved in and responsible for the business processes with up-to-date status information regarding the current business processes. In the left hand window, Fig. 36 shows the business process defined at the engineering level. In the right hand window, it depicts the deduced instance process. Although the structure of the processes is identical, there are still differences between the two. The folder symbol in the function „maintain puchasing data of material masters“ indicates that this function is currently being worked upon. The previous functions have already been executed. Thus, the roles of the organizational units are known as well as the roles of each individual employee. Functions for which only the roles of the employees are involved and which have not yet been initiated are color-coded.

In addition to the processing status, current process times and process costs can be shown ad hoc. This provides the persons responsible for the business process with transparent information for answering customers' questions and manipulating the remainder of the process if necessary.

D.II.2 Scheduling and Capacity Control

Business processes lead to transaction sequences in accordance with network diagram techniques. When expected or planned scheduling values are allocated to the functions, it is possible to compute the event network by known measures of the network diagram technique, such as latest or earliest starting time or finishing time of events, and thus compute the entire process.

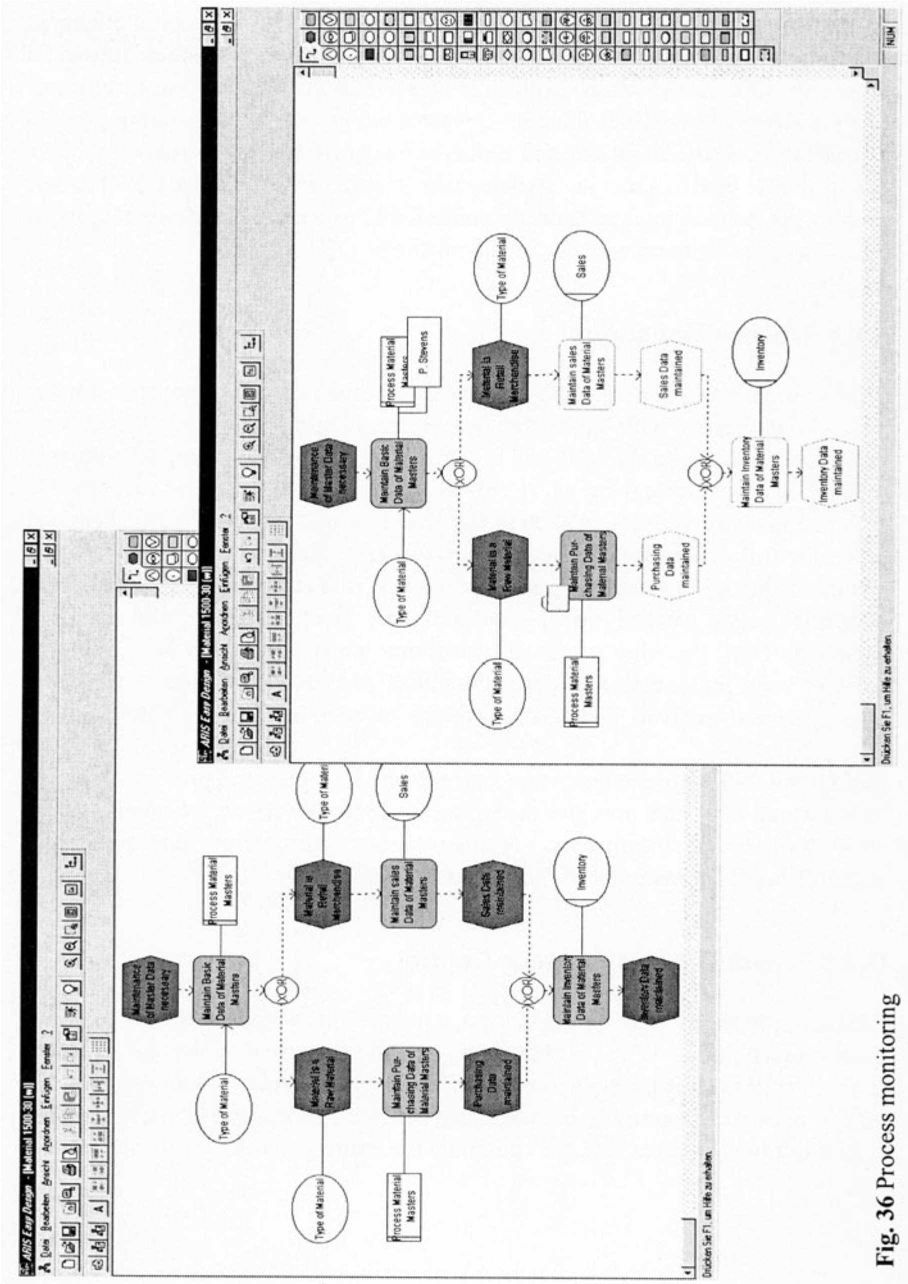


Fig. 36 Process monitoring

Fig. 37 depicts the result of such a calculation in a Gantt diagram. This example refers to the procedural model for the software implementation process. One line in the Gantt diagram is created for each function of the process model. The length of each bar indicates the duration of the function. The duration of the

functions is calculated according to the control flow and according to the intervals allocated to the functions (for example, the function target concept B begins one week after the function target concept A). By linking organizational units and machine resources to the functions, capacities can be illustrated as well. Every line of the Gantt diagram represents a resource and shows its load for that period of time. Fig. 38 depicts such an example. Capacity load overviews are shown as bar charts.

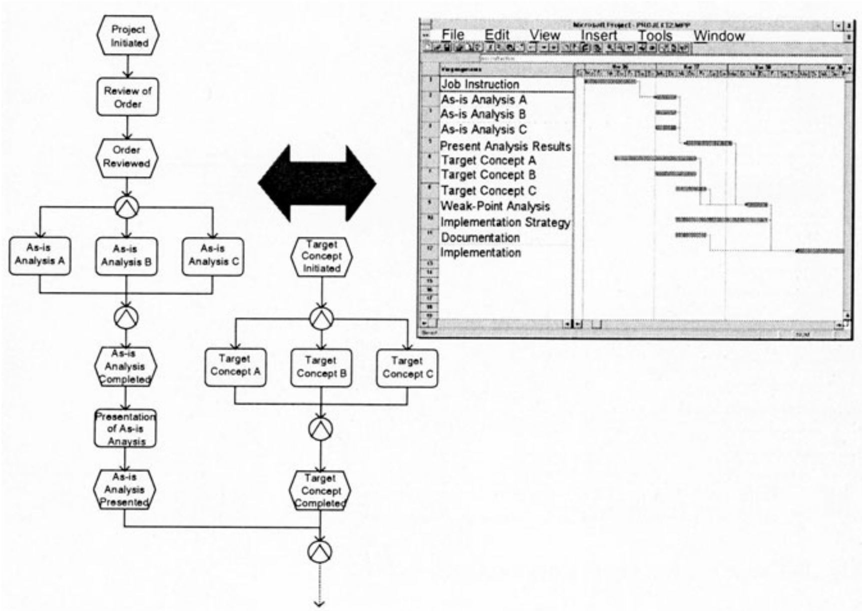


Fig. 37 Scheduling a business process
(procedural model for software implementation)

Production planning and control systems, which are becoming increasingly popular, provide a wide range of algorithmic planning aids. When things become tight, algorithms can plan overtime and additional shifts, transfer operations to utilize spare capacities, or heuristically plan operations of various orders (processes) according to assigned priority values.

Capacity and planning management is increasingly being used beyond production tasks as well. The benefits they provide the persons in charge of the business processes go way beyond its traditional arena. Considering the expensive resources in operating rooms, it is useful even in hospitals. In retail, especially for global operations operating in countries where laws regarding the flexibility of opening hours are being liberalized, work management is becoming an increasingly important issue. This is even the case in public services, where seasonal variations in the amount of work (such as processing

annual wage tax recomputations) make the usage of scheduling and capacity control tools advisable for evening out employees' work loads and keeping processing times down to a customer friendly level.

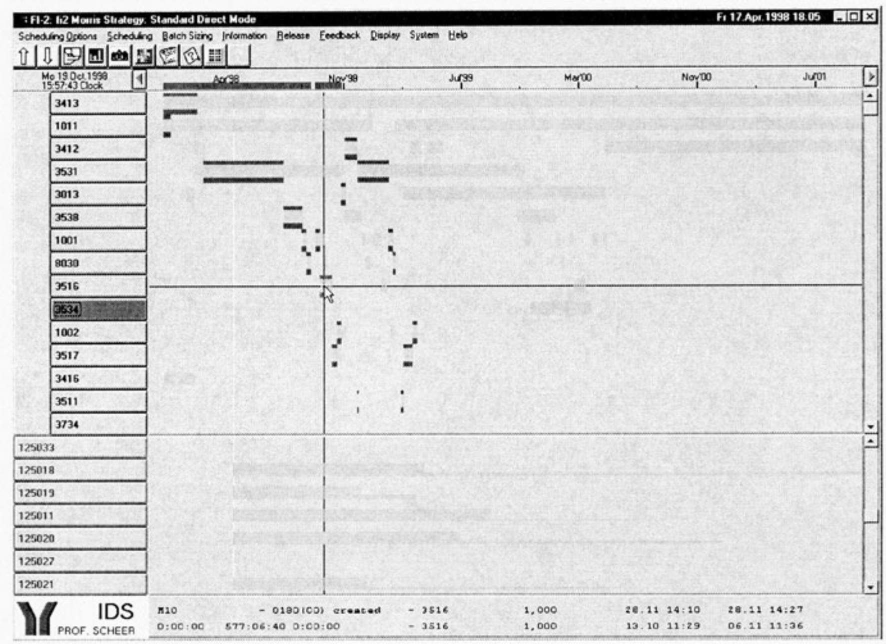


Fig. 38 Capacity planning of a business process

D.II.3 Executive Information Systems (EIS)

Just because the amount of available information has increased does not automatically mean that the quality of business decisions is going to improve. On the contrary, information overload often has the opposite effect. Reams of dead data tend to blind decision makers who „cannot see the forest for the trees". Crucial elements for decision making should therefore be filtered according to person-specific and task-relevant requirements. Executive information systems (EIS) deliver, providing management with aggregated data on corporate and external issues, enabling decision makers to monitor, analyze and plan business processes. Ease of use makes them suitable for management, i.e., the business process owner.

Key EIS characteristics are as follows *(see Back-Hock, Executive-Information-System-Generatoren und -Anwendungen 1991, pp. 40-41)*:

They
- Enable automatic merging of data from various sources,
- Are easy to use,
- Query information from various perspectives and levels of aggregation (drill down functionality),
- Provide aggregation functionality,
- Provide user oriented reporting functionality, including graphical evaluations,
- Feature additional functions (printing, data management, e-mail with reports and comments).

Data warehouses provide an IS concept for executive information systems *(see Inmon, Data Warehouse 1993).* Data warehouses rigidly separate operational and DSS (decision support systems) data and systems. Whereas current data is stored only in operational databases, historical data is stored in data warehouses. When comprehensive queries are made, accessing data warehouse databases directly ensures rapid response times. Another benefit is that the performance of operational systems is not slowed in any way. The disadvantages of data warehouses are increased effort due to redundant data storage, and the need for data integration after data warehouse updates. Another drawback is the lack of data integrity between data warehouse updates *(see Scheer, Data Warehouse 1996, p. 75).*

The special requirements that executive information systems demand of database systems are characteristic of data warehouses. Their table structure is capable of modeling two dimensions, say, „time“ and „products“. More complex data analysis, such as

- Enabling statistical analysis across various databases,
- Executing complex statistical calculations or
- Creating reports with data resulting from multiple database areas

would exceed the basic manipulation functionality of which relational database systems are capable *(see Engels, OLAP 1995, p. 99).*

Codd, the father of the relational database model, also proposed OLAP (online analytical processing) databases which enable complex data analysis operations by storing multidimensional measure information. OLAP lets decision makers do autonomous on-line analysis *(see Codd, OLAP 1993; Jahnke/Groffmann/Kruppa, OLAP 1996).* Thus, we should regard OLAP as a subaspect of a comprehensive data warehouse concept.
One of the characteristics of OLAP is that it enables information to be analyzed from various perspectives. This makes it possible, for example, to analyze costs from the perspective of certain regions, time frames and business processes. OLAP sorts values of various reference objects (regions, time, processes, etc.)

in dimensions parallel to the axes of multidimensional cubes. Here is an example of an OLAP ad hoc query: „What are the costs across every process in a certain region in months 1 to 3?“ Evaluating the same data from another perspective could lead to this query: „What are the costs for a certain process in every region in the past quarter?“ Business owners can use EIS to aggregate information resulting from current business processes. EIS evaluations can be preconfigured in accordance with process model logic, especially in accordance with the data view. Combinations of the following segmentation structures are frequently used as reporting or evaluation dimensions *(see Muksch/Holthuis/Reiser, Data Warehouse-Konzept 1996, p. 424; Zell, Führungsinformationssysteme 1997, p. 293):*

- Organizational structures: Rigidly group organization plans by legal units, business, process and function areas, departments, cost centers, etc.,
- Product structures: Sort by items, product groups, types, etc.,
- Regional structures: Sort by countries, states, areas, regions, etc.,
- Customers / sales types: Sort by customer groups, customer types, sales types, sales channels,
- Time structures: Sort by report frequency (monthly, quarterly, annually) and report period (month, quarter, year),
- Business characteristics and measures: For example, sort by revenue, contribution, profit, process costs, etc. *(see Reichmann, Controlling mit Kennzahlen 1997),*
- Data categories: Sort by forecast, target, actual, variances.

The results of level I, particularly benchmarking and simulation studies, can also be used as sources for EIS. Results from process monitoring, and scheduling and capacity control of level II, based on information on the execution of processes in level III, are also key sources.

Intelligent query strategies (data mining) enable process owners to precisely navigate to processes relevant for the query. Data mining concepts aim to identify patterns in comprehensive and structured data that would otherwise be hard to make out, and to present them to the end user as relevant information *(see Hagedorn/Bissantz/Mertens, Data Mining 1997; also see Petersohn, Klassifikation bei Entscheidungsproblemen 1997).*

We will skip reviewing evaluations that basically only document the planned procedure of business processes.

Fig. 39 depicts exception reporting in business process management *(see Kraemer, Kostenmanagement 1993).* This enables a quick and aggregated overview of problematic areas in the enterprise. Shaded areas visualize for which processes immediate correction measures (dark gray) are recommended, which processes should be analyzed further (light gray) and which processes do

not require further analysis (white). Detailed analysis of the identified processes can provide pointers for additional process optimization.

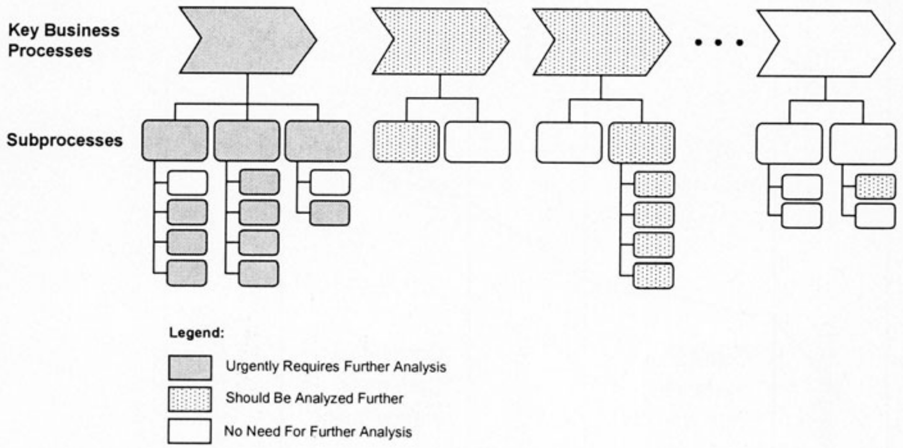


Fig. 39 Exception reporting in business process management

D.II.4 Continuous Process Improvement – Adaptive Business Process Engineering

Engineering business processes is not a one-time event within a company, but rather a continuous task of the person(s) responsible for business processes. Kaizen, the Japanese management paradigm *(translated as "slow, never ending optimization", see Sebestyén, Management-„Geheimnis" Kaizen 1994, p. 17)* stresses the need for constant adaptation and optimization of business processes.

In addition to continuous or evolutionary business process improvement, Hammer/Champy's BPR concept *(see Hammer/Champy, Business Reengineering 1995)* takes a more revolutionary approach. With BPR, the goal of enterprises is to engineer their processes as if they were starting from scratch.

Both approaches have their merits. If a specific situation arises in which an enterprise has the opportunity to do some fundamental rethinking regarding its innovative structures, this can result in a BPR project. But even after completion, processes stay in motion: New organization forms appear, new best practice cases become available as reference models, new technologies are invented or expertise is gained with processes only recently implemented. All of this leads to new process adaptation. The common term „turbulent environment" is indeed a realistic way to describe the adaptation requirements demanded of an enterprise.

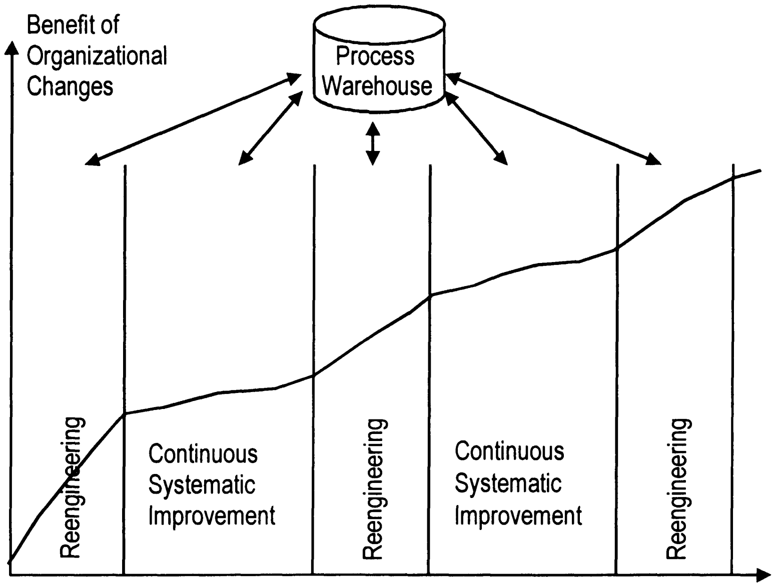


Fig. 40 Reengineering and continuous improvement

Planning and controlling business processes visualize the various reasons for reengineering, linking the engineering and controlling levels with one another, as shown in the HOBE feedback loop (see Fig. 24). Fig. 40 *(in accordance with Imai, Kaizen 1992, p. 51)* depicts fluctuations between large reengineering phases and continuous improvement.

Frequently, companies take advantage of implementing or migrating large IT concepts (for example, implementing an integrated standard software solution) to do major reengineering. This avoids applying the new technology to old processes. Leaner organizations also serve to streamline software implementation. Moreover, companies can take advantage of this phase of better employee motivation for improving reengineering process. For example, the following process changes could become necessary for process improvement:

- Modifying a function procedure,
- Consolidating several functions,
- Modifying the control flow,
- Modifying organizational responsibilities,
- Modifying the data in use,
- Modifying IT systems.

Business processes are considered robust when changes in the corporate environment require little or no modification of the business process. If modifications become necessary, the costs of these modifications depend on whether the process is easy or difficult to adapt. It is obviously important for

business processes to be robust and adaptable, although it is difficult to quantify these measures.
Business processes should be clearly documented during reengineering phases and continuous process optimization. This is a prerequisite for rating them, as shown in process engineering. Fig. 40 illustrates this by the designation „process warehouse". This is where corporate organizational knowledge regarding processes, and information on reference processes are stored. As a foundation for future organizational developments, current, legacy and even future process models can be captured here as well.

Fig. 41 illustrates such a path for developing models. If past models are stored as well, models can be compared and rated in order to appraise the success or failure of reengineering measures.

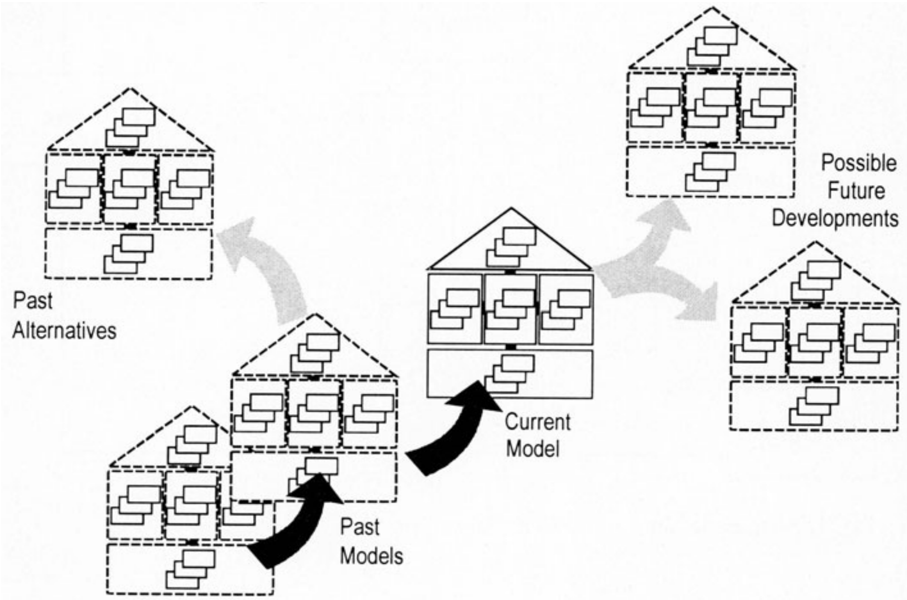


Fig. 41 Various types of models
(in accordance with Allweyer, Adaptive Geschäftsprozesse 1997, p. 186)

Version control is necessary for storing key model versions. Fig. 42 depicts the necessary meta model. By linking them with the class TIME, models and the individual model elements of which models are comprised are time-stamped. The respective data is stored in the data objects MODEL VERSION or MODEL ELEMENT VERSION. Model versions are comprised of the model element versions allocated to them. The term „model element" represents a variety of aspects such as functions, data, organizational units, output and its interrelationships, from which real-world process models can be compounded.

By using *:* associations between the various model and model element versions, it is also possible to allocate several versions of the same model element to one model version. This prevents small changes in model elements from automatically leading to new model versions. From a corporate point of view, models consist of several hundred or even thousands of elements, making version control a complex task. On the other hand, version control enables the person(s) responsible for the business process to manage business processes transparently. Version control is also part of corporate organizational memory.

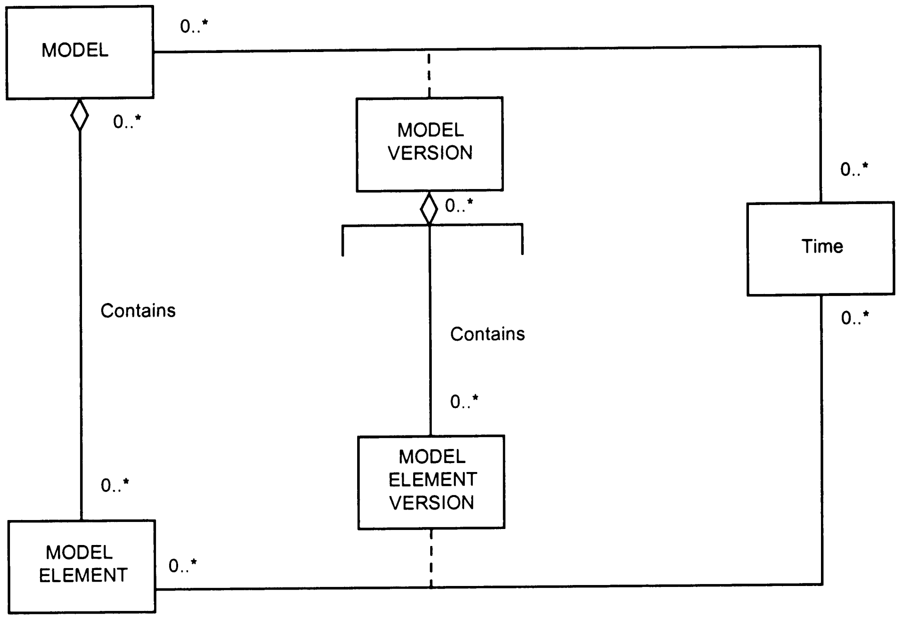


Fig. 42 Meta model for controlling model versions

D.III Workflow Control

Business process engineering and business process planning levels, respectively, are geared to business oriented managers. Workflow control converts business processes into IT tools.

Generally, it is not possible to administer an entire business process with one application software system. Very often, a variety of systems for sales, purchasing, manufacturing or accounting is necessary. Even integrated standard application packages have gaps which have to be filled by custom systems or standard applications from other vendors. None of these systems is individually capable of determining the status of the entire process (for example, every processing state of a particular order). It therefore makes sense to allocate the responsibility for comprehensive process control to an explicit system level rather than distributing it across several systems. This level is called „workflow".

Workflow systems pass the objects (documents) to be processed from one work place to the next. Ideally, they do this electronically, from the computer system of one workplace to the next operation step's system. This requires a detailed description of the procedure, customized for the individual process type, and of the respective employee.

The document flow in Fig. 24 is characterized by a „folder", containing electronic references on the function components to be started, and the data necessary for processing, passed from one workplace to the next.

Fig. 43 shows how a process which is defined at the engineering level turns into a real-world process at the execution level. Instead of listing general names for organizational units, we now have actual employees. Instead of general order designations, orders refer to actual customers. After a particular operation has been completed, the workflow system takes the document from the clerk's electronic outbox and passes it to the electronic inbox of the next clerk. If several clerks are available, the process can be placed into several inboxes. As soon as one clerk starts processing it, the process is deleted from the other inboxes *(see Hagemeyer/Rolles/Schmidt/Scheer, Arbeitsverteilungsverfahren 1998).* Fig. 44 depicts the screen of the workflow system, including icons for intrays, outtray and clipboards.

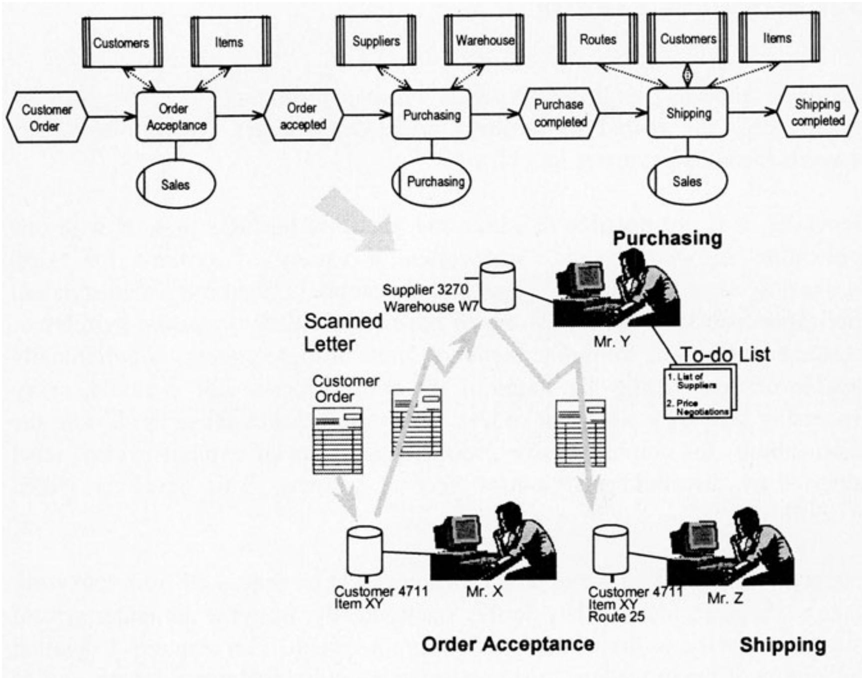


Fig. 43 From the business process model to the real-world procedure

Workflow systems contain information on the processing status, execution times and the users of every business process, returning data for cost and time evaluations and providing information for process monitoring. This is why workflow systems are the foundation for process controlling at level II.

Process illustrations of workflow systems also act as user menus. This makes the organizational background of the business processes more understandable. EPC illustrations, regarding process monitoring (see Fig. 36, right hand window) as discussed, are key in this context. Particulars include defining individual employees and selecting a certain path from the various alternatives in the general business process description. This lets users see their role within the process, who their „predecessors“ and who their „successors“ are. They can also see that, as shown in this example, only the left branch of the business process is relevant for them because the control flow in the right branch is deleted. Since a particular user is not yet been named for the next activity, only the department name is listed. Due to the current capacity load, the user in the next operational step is not named until after completion of the activity.

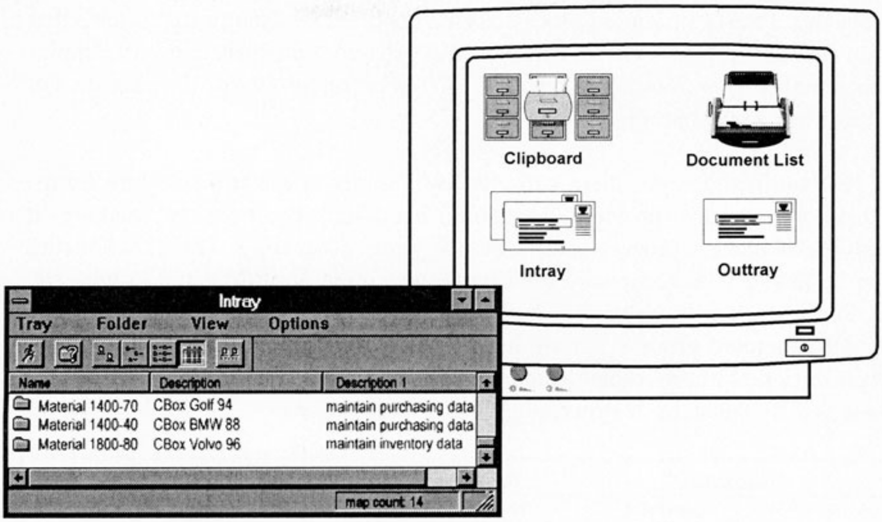


Fig. 44 User view of workflow control using clipboards

In process workflow, we distinguish between processes with a clearly defined process structure and those with only a preliminarily defined procedure sequence. In many operational and repetitive procedures, as in order processing or loan processing, the functions, their sequence, process branches and organizational units are predetermined. Processes are well-structured and can be described by the EPC method, for example.

Other processes can only be partially described because the functions are not known until the actual processing, the sequence of the processing steps is determined ad hoc, and organizational units become apparent from ad hoc requirements. These processes are regarded as poorly structured and may only be modeled in part. For example, their functions can only be listed in to-do lists. The exact sequence is determined by teams during the execution and is the responsibility of the person(s) executing it. In ad hoc workflows, employees determine their own „successors".

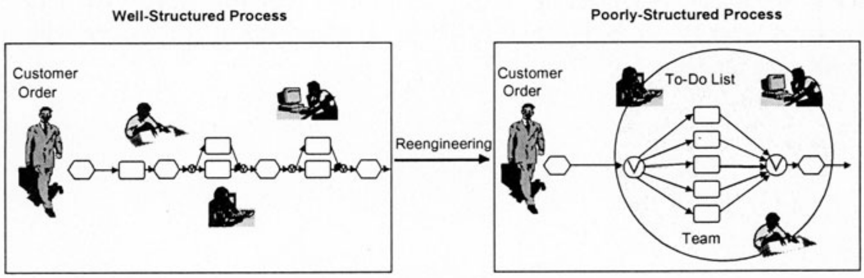


Fig. 45 A process structure before and after the implementation of the team concept

At first glance, workflow systems are only suitable for controlling well-defined processes. Poorly structured processes are supported by groupware systems that only feature e-mail, video conferencing, shared conferencing and similar functionality *(see Schwabe/Krcmar, CSCW-Werkzeuge 1996)*, but that do not require any logical process skills.

In real-world situations, there will always be a mix of the two structure forms. Workflow systems provide „exception handling" functionality, making it possible to change process control ad hoc while processing. This functionality can be linked with groupware tools, complementing workflow and groupware. In the future, it will probably even be possible to merge the two. Fig. 45 shows a well-structured process turning into a poorly structured process after a team organization has been implemented. Here, only a to-do list is prescribed. Fig. 46 depicts different steps of structuring workflow processes.

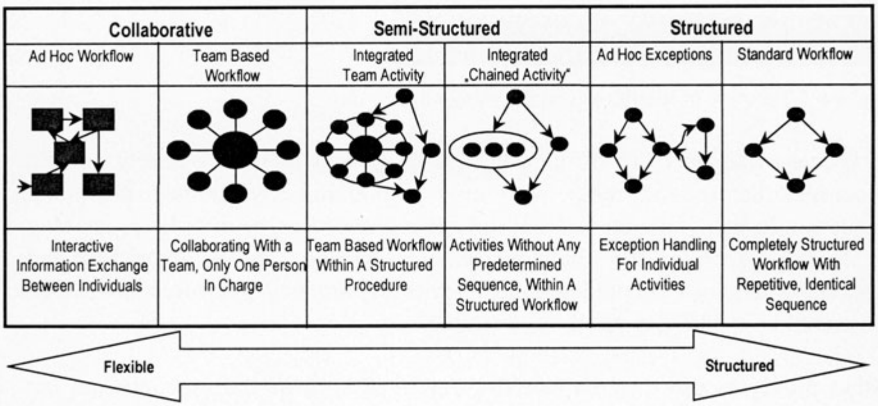


Fig. 46 Various degrees of structuring workflow processes *(see Vos, Groupware 1997, p. 40)*

Thanks to the development of interface standards by the workflow management coalition (WfMC), different workflow systems can now be linked by application programming interfaces (APIs). Currently, interface standards for process modeling (engineering level I), process control (level II) and application level IV are being studied. These are classified in accordance with the reference model in Fig. 47.

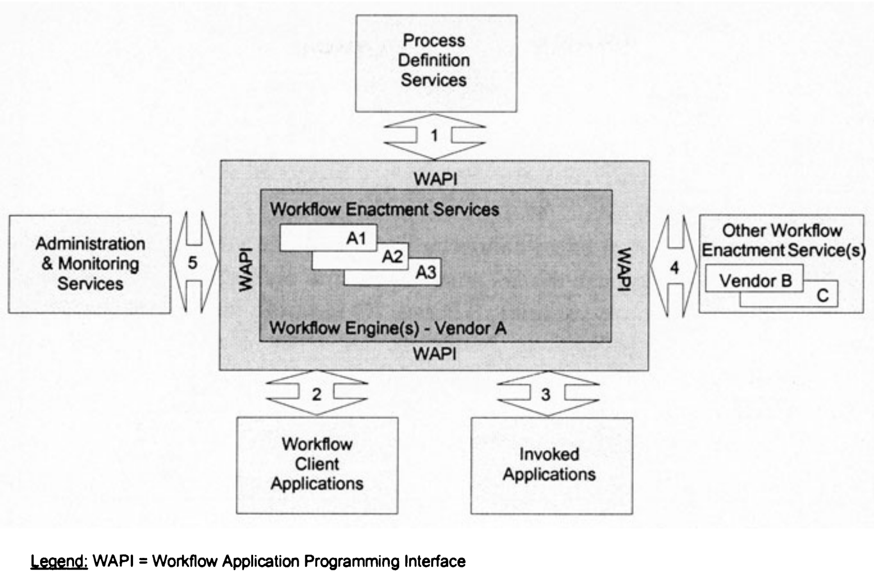


Fig. 47 Reference model of the workflow management coalition
(from Hollingsworth, Workflow Reference Model 1995)

Standards such as these are generally minimal solutions because they agree on the smallest common denominator of the respective parties. Nonetheless, they are valuable aids for the further development of an open, process oriented software architecture. For real-world applications, however, the most successful method is still to develop individual interfaces between the respective systems.

In order to directly provide workflow systems with data from process modeling level I, each instance must be generated in accordance with the model templates. Administration of the instance data must be possible as well. Instance models should be modified in the event of exception handling, too.

Instances are usually maintained by the application systems. Workflow systems, however, are overriding concepts designed for controlling purposes, independent of any particular application. Due to the fact that instance data, such as the starting and finishing data of an event, is captured without any application oriented additions („start order processing“ or „start manufacturing process“), it is maintained by models located at level I of the ARIS repository.

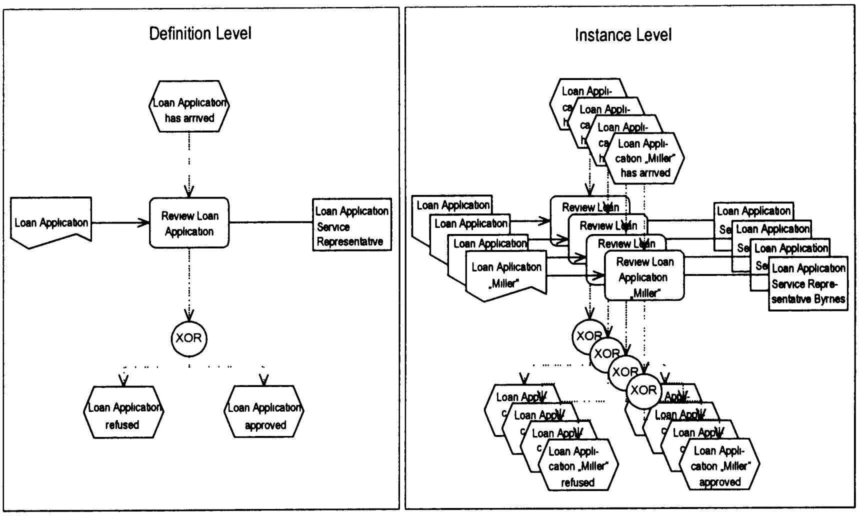


Fig. 48 Types and instances of a business process

Fig. 48 illustrates an example for an excerpt of a business process for processing a loan at the type and instance levels. At the meta level, an instance object is allocated to every type object by a 1:* association (see Fig. 49). In the „modeling levels" chapter, we will discuss in detail how instance models are embedded in the ARIS concept (particularly in the meta and meta² structure of the repository).

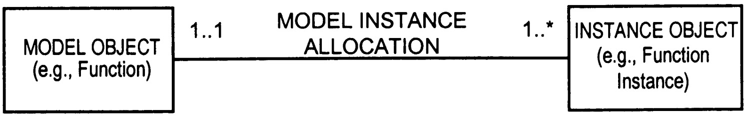


Fig. 49 Association between a type object and an instance object at the meta level

D.IV Application Systems

Workflow systems start applications at the HOBE level by CALL commands, making it unnecessary for users to be familiar with the application system. As soon as the CALL command activates a case to be processed from the intray, the workflow system starts the application system and the screen appears, along with the data necessary for this process. The data is populated by database

comments stored in the electronic event folder, along with comments regarding the processing programs to be used.

In order to control programs with a workflow program, the granularity of the modules must be sufficiently fine. Also, the modules must be capable of being externally accessed by the workflow system.

Initially, we will touch upon the various possibilities of controlling traditional standard software by processes. Then, we will discuss object oriented approaches, which are becoming increasingly important, and which lead to componentware. Framework concepts are embedded in the HOBE concept, resulting in architecture information. This also makes software components reusable and thus outlines the focus of this approach.

D.IV.1 Traditional Standard Software Solutions

Traditional standard software solutions are transaction driven, integrated business application systems. Frequently consisting of integrated programs and characterized by programmed process control, these solutions are only partially well-suited for the workflow driven HOBE concept.

Provided the systems are componentized and provide an interface for remote procedure calls (RPCs), however, workflow systems are capable of starting these modules or transactions. For example, SAP R/3 has an interface that utilizes SAP's remote function call (RFC). When independent workflow systems are used to control standard software, the standard software is divided into processing functionality and the control flow passed on by the workflow. This means that the manufacturer is no longer responsible for the integrity of the entire solution. In order to pass this flexible approach on to their architecture by workflow systems, standard software solutions use custom workflow systems for process control. Unfortunately, these workflow systems are often closely linked with the system, preventing them from fulfilling the requirements of a process control solution based on heterogeneous systems.

Standard software solutions can be configured by these models if documented by semantic models and if the models are linked with the system repository.

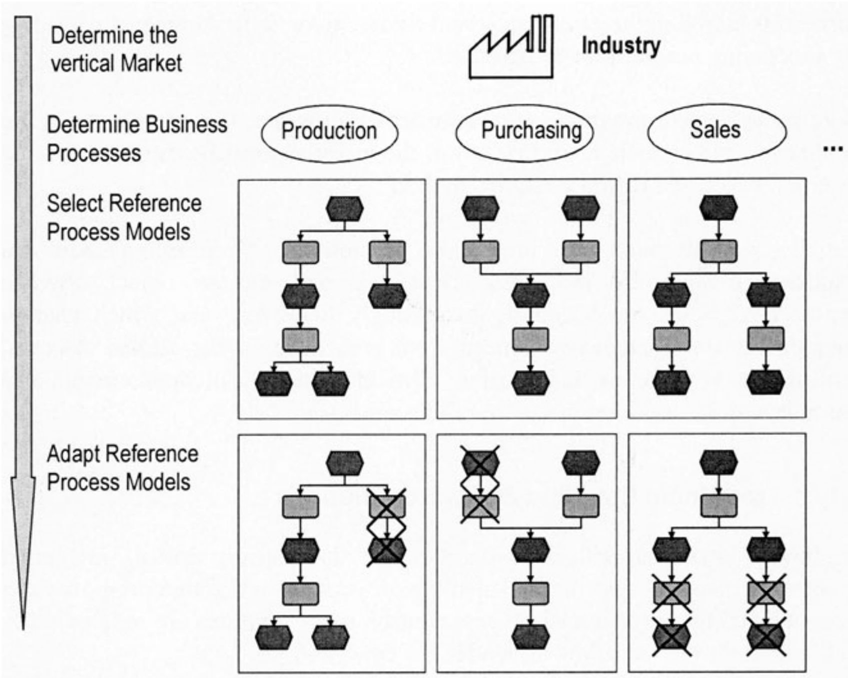


Fig. 50 Individualizing reference models

Redlining hides unnecessary functions of a business process provided by a software application. Fig. 50 depicts the schematic procedure of SAP R/3, supported by a model. First, the reference model of the respective business line is determined such as retail, banks, industry, etc. In our example, we have selected the vertical market customer „industry". Within the vertical market solution, specific business processes are selected, in our example, production, purchasing and sales. In order to obtain customer-specific business processes, unnecessary functions and events dependent upon them are deleted from the processes, illustrated as EPCs. Redlining stores certain rules so business plausibilities and dependencies can be observed. For example, if the function „storing" is canceled, the Function „releasing from stock" must be deleted as well.

Customizing information is directly passed through to the system configuration management.

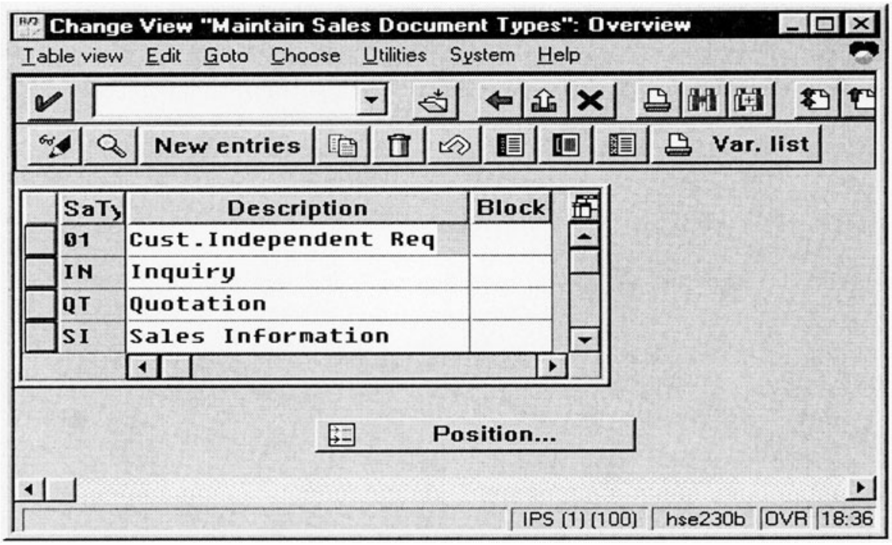
Each business function offers additional parameter options. For example, a function „create purchase order" can refer to a consignment order, a third-party order, a warehouse order or a sales office order. Obviously, this order must be specified in more detail. SAP R/3's implementation management guide (IMG)

provides additional computer-aided help. As in process models, other model types such as data models, function models and organization models can be used for configuring standard software. For example, information objects can be deleted from or added to the standard software data model. Attributes can be deleted or their length can be modified. This information is also passed on to the system, automatically customizing the screens. For further examples, see *Scheer, ARIS - Business Process Modeling 1998.*

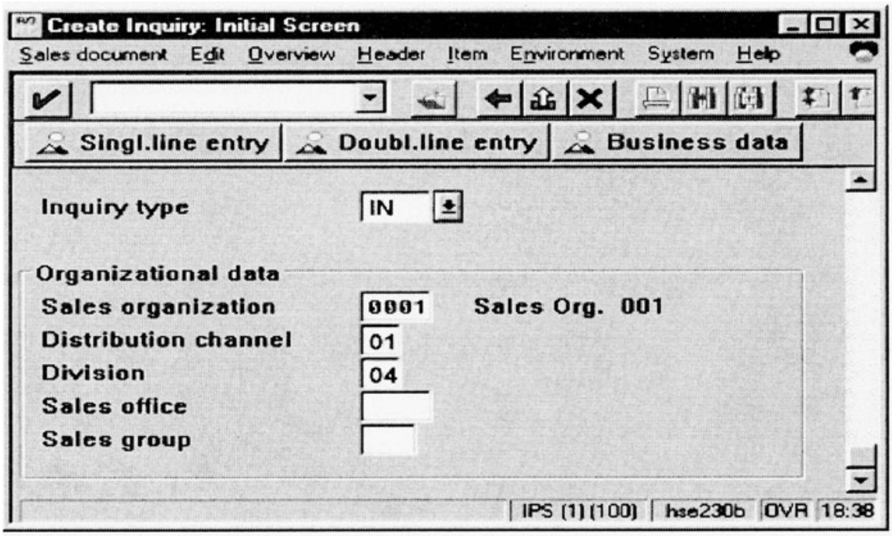
Direct interaction between the modeling tool, the semantic models and the application system modifies the implementation strategy of standard software. Rigid phase concepts used to be customary, with as-is analysis pinpointing any potential weak spots. Then standard software independent target concepts were developed and implemented, using customizing techniques. Now, these phases are increasingly executed simultaneously and interactively. This lets business and implementation staff work in close collaboration, leading to simultaneous addressing of business and implementation issues. Fig. 51 shows an example. For clarification purposes, the four windows are shown separately. In a real-world application, these windows would be displayed on one screen, providing the user with all the information at once.

The upper right hand window shows the excerpt of the business process model in the ARIS modeling tool, illustrating the part of the standard software process that can be hidden. An additional process branch, not contained in the standard software, must be added.

The function „create inquiry" asks users which screen is being used in SAP R/3. Using the process model as a modeling tool and by clicking on the function (or starting a command), users can seamlessly invoke SAP R/3. This screen is shown at the bottom left.

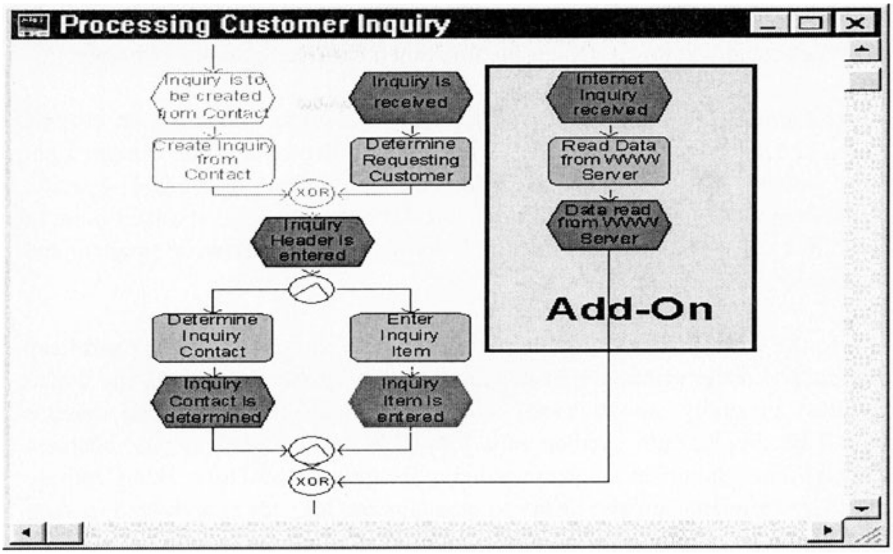


Customizing: Maintain Sales Document Types



Function „Create Inquiry“

Fig. 51 Interactive business process engineering and customizing of standard software



Process Model

Process: Processing Customer Inquiry						
Function	As-is/Target	Unresolved Issues	Inter-face	In Charge	Date	Effort
1. Determine Ordering Customer	From now on, ord-ering Customers will be queried in Accordance with ISP Country Codes	CPD Customer necessary	Customer Master Data (internal)	C. Jones	May 29	Standard
2. Determine Inquiry Contact	Define third-party as new Partner Type in customi-zed Version	None	Customer Master Data (internal)	P. Miller	May 29	Standard
3. Enter Inquiry Item	Use AFN Item Type as Standard	None	Customer Master Data (internal)	P. Miller C. Jones	May 30	Standard

Documentation of Results

The IMG customizing tool is activated for customizing the function. In the upper left hand window it depicts the function parameters that are available.

Using the modeling tool, results of discussions, parameter decisions, unresolved issues and the like are stored in the function, as depicted in the bottom right hand window. This enables detailed documentation of business and IT-specific business process engineering. This documentation can be used at a later point in time for clarifying questions, using this know-how for follow-up projects and for monitoring the project.

Due to the large effort required to maintain highly integrated systems and keep training and documentation manuals for the entire system up-to-date, and due to vendors' generally slow release policies for major releases, these systems should be divided into smaller units (modules, components, agents, business objects). For components, release policies are quite satisfactory. Being loosely linked, components are also easier to maintain and it is not as awkward to keep their documentation up-to-date, either.

Splitting up software systems into modules is not a new concept. Modules provide users with precisely defined interfaces, their sole means of exchanging information. Users are not burdened with the implementation of the module (information hiding).

The modular concept does have one disadvantage, however, in that there are multiple cases where it can only be used in exactly the same way. As soon as a module for certain applications is modified, the program code should be manipulated. This can even lead to different versions. This tends to clutter modular concepts and limits their reusability.

On the other hand, object oriented concepts are defined at the type level, making it possible to create subtypes of a certain object type when changes are made -- without having to change the original object type. Subtypes only define the deviation from the original object type. This enhances the flexibility of the system design as well as the reusability of the design objects.

With industry dividing monolithic software systems into smaller and smaller pieces, this is dovetailing with the increasing possibilities of object oriented concepts. Which leads us to the buzzword „componentware", the way real-world applications are actually implemented.

D.IV.2 Componentware

The main idea of componentware is to assemble software systems from individual standard components developed by various vendors. These components are loosely connected by message interchanges. Program development shifts from programming -- to designing -- solutions and assembling the components. The concept of using components is closely linked with the basic principles of the object oriented concept. Object oriented concepts are not new, although their role in real-world applications has been increasing in the past few years. Today, new application software is generally developed using object oriented technologies, with the top-down approach of dividing traditional standard software into object structures coming full circle with the bottom-up approach of developing new systems.

D.IV.2.1 Objects

Object orientation is based on the concept of encapsulating objects with their respective data descriptions and the methods (functions) to be applied to them. Users can activate methods using messages, enabling data access. The actual method implementation is hidden from the user.

Systems are developed at the type level, i.e., similar objects are grouped into classes. In this work, we do not distinguish between the designations „object" (instance level) and „classes" (type level), but rather simply refer to objects. The context makes clear which term is being referred to. Another characteristic of objects is their inheritance functionality. This enables methods and attributes of overriding classes to be inherited by the subordinate classes by means of generalization / specialization. Inheritance supports the basic reusability principle.

Obviously, the characteristics of object oriented methods could be described in much more detail, but this preliminary sketch shall suffice for this work. In Fig. 52, referring to the „order processing" example in chapter B.I., objects, along with their names, attributes and methods, are depicted. This figure also illustrates the message interchange, including the method of the respective target object, the data to be transferred and the data contained in the reply. As opposed to Fig. 5 showing the information flow, we now differentiate the functions triggering the data exchange. The message flow resulting from the business process control flow should not necessarily be used because the sequence of the function execution within the objects is not listed. We want to call to mind that business processes have multiple flows such as function, output, information and organization flows, respectively. Workflow control stresses the function flow, whereas the object oriented concept focuses on the message flow between the information objects.

Various programming languages (Java, C++, Smalltalk, etc.) are based on object orientation. When developing programs, object libraries make it possible to reuse tested objects.

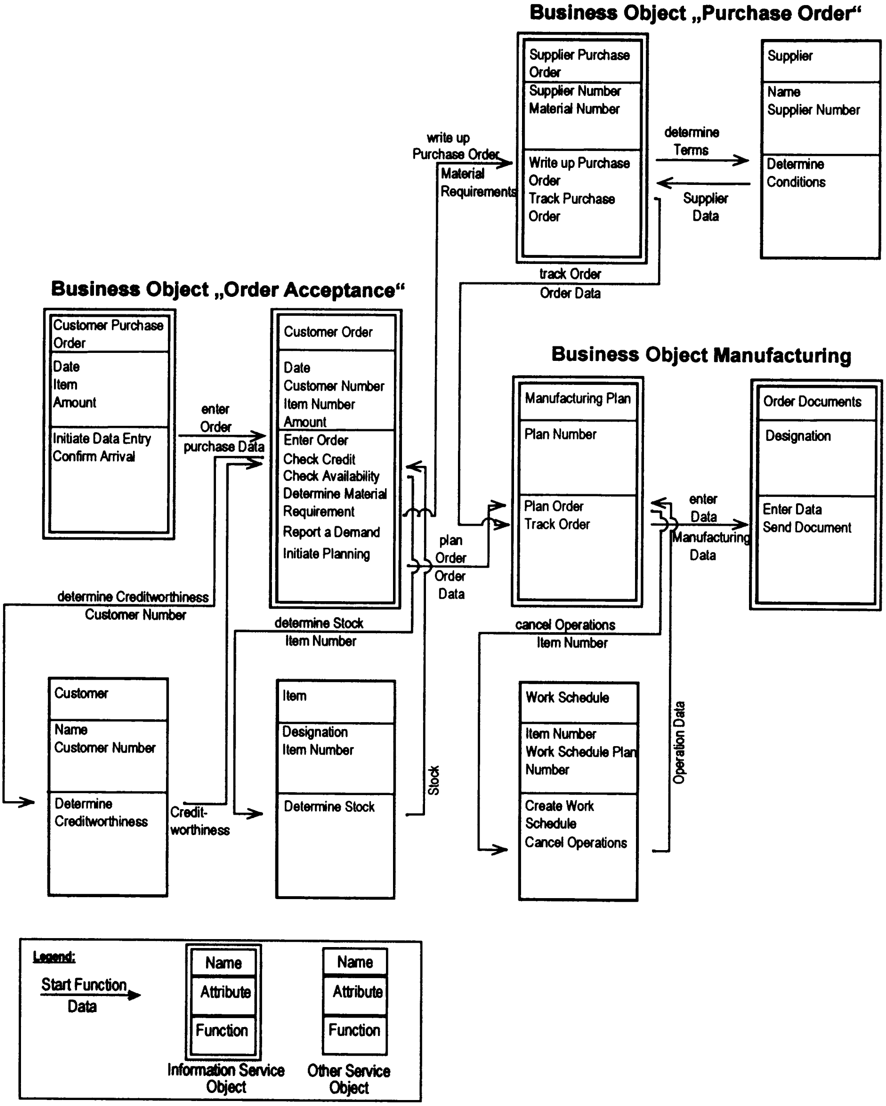


Fig. 52 Object oriented illustration of the example „order processing"

Adhering to these basic principles by the use of an object oriented programming language, however, by no means guarantees an increase in performance, as opposed to traditional module concepts *(see Pree, Komponentenbasierte*

Softwareentwicklung 1997, p. 5). One of the most common reasons for comprehensive systems to become cluttered is when their objects' granularity is too high. Moreover, when individual objects are this finely granulated, it is difficult to start them from a sensible workflow control. This is why objects are frequently grouped in larger units, so-called „business objects", which include application information on the interaction of the internal objects.

D.IV.2.2 Business Objects

When developing software „assembly style", the granularity of object oriented objects (stemming from programming) is too fine. What we need is logical building blocks with a greater degree of „coarseness". For example, the object oriented UML concept recommends so-called „packages". This application oriented view, which defines coarser objects in accordance with their application functionality, is easier to understand when we call it a „business object". Business objects comprise business processes, along with the respective data and the functions to be applied to them. Thus, business objects contain multiple object oriented objects, as discussed. Their characteristics like encapsulation, reusability (resulting from inheritance) and loose linkage (resulting from message interchange) are derived from the object oriented concept *(regarding business objects, see Casanave, Business-Object Architectures and Standards 1997; Fingar, Blueprint for Business Objects 1996; Fingar/Read/Stickeleather, Next Generation Computing 1996; Burt, OMG BOMSIG Survey 1995).*

In Fig. 52, the order processing example contains three business objects -- for order acceptance, purchase order and manufacturing. In Fig. 53, the business objects are illustrated in a characteristic manner. The interfaces of the various objects, started externally, are passed on to the outermost business object, illustrated by dots along the outer border of the object. Interfaces only used within the business object, however, remain within the borders of the inner objects. Business objects are not characterized by any pre-determined granularity. Rather, granularity is chosen according to sensible application oriented work units, for example, depending how close links with the respective contents and organizational structures are, how strong internal communication is (along with weak external communication), according to the degree of reusability and performance.

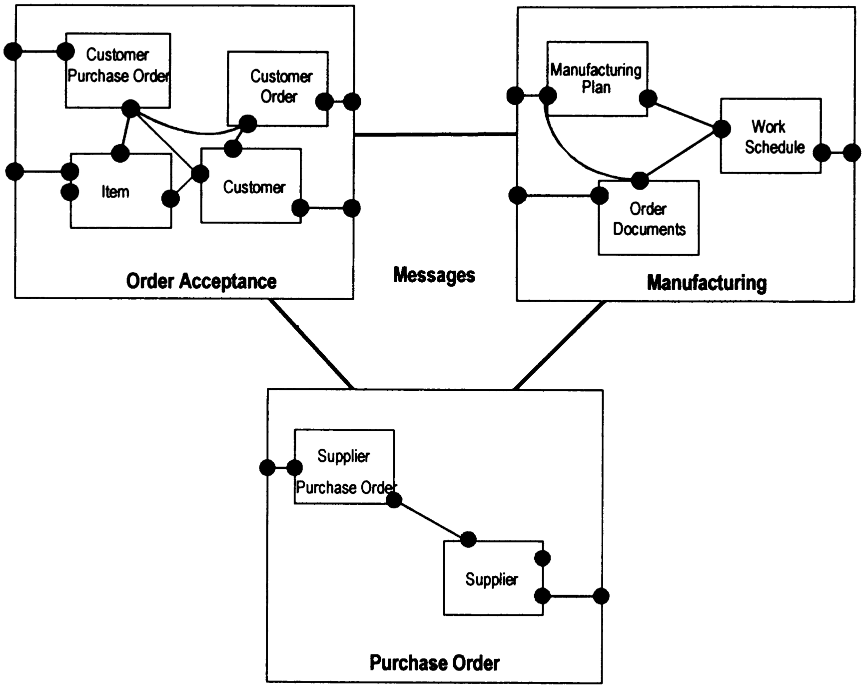


Fig. 53 Business objects in the example „order processing"

Business objects do not necessarily have to be created from individual, rigidly object oriented objects, but rather can be made up of conventional program components when an existing system is broken up in a top-down method. The main prerequisite is that the business object conforms to object oriented principles, as does SAP R/3 for example. To date, 170 business objects are available from SAP *(see SAP, White Paper Business Objects 1997),* stored in the business object repository (BOR). Their internal structure is written in ABAP 4GL which is not (yet) object oriented.

D.IV.2.3 Java Applets

Componentware is also supported by software concepts utilizing extremely simple hardware clients, so-called network computers. Applications are stored in a network (Internet or intranets) and are started by the client as need may be. Java's functionality is especially valuable in this respect. Thanks to Java applets and popular Web browsers as front ends, platform independent processing is now possible.

In order to develop a Java applet, source code is developed in a Java development environment on any system. The completed program, so-called bytecode, is then compiled on the development system rather than in an

executable program. Bytecode is platform independent and must be interpreted once again before being executed, if it is to correspond with the technical requirements of the client system. This is done by means of the Java virtual machine (JVM) after downloading. JVM adapts the bytecode to the various client requirements.

Java can be used at three different levels. Commands can be entered directly into the text of an HTML page, as JavaScript source code. The command code is only partially identical with Java, although both languages are based on C++. However, only small applications can be implemented with JavaScript, for example when reviewing the validity of values or displaying a banner in the browser status line. JavaScript does not permit direct communication between the client and the server.

In phase two, the applet bytecode is passed to the client, in addition to the loaded HTML page. The bytecode is verified and interpreted on the client, after which the applet is executed. However, applets can only be started within the browser environment. They are not executable without it. Within the applets, however, further communication between server and client is possible. Applets can upload documents from the server and process them before they are sent back to the server for the conclusion of the process. Applications can be run independently of the browser environment. All they need is a Java Virtual Machine for transferring the bytecode. Applications enable the development of independent applications, whose building blocks are downloaded to the client via a network, whenever necessary. Both applets and applications are capable of supporting componentware. There is no need to install and maintain unnecessary modules. Systems are assembled individually, in accordance with customer specifications.

This concept can be included in the HOBE framework (see Fig. 54). The processes controlled at level III by the workflow system are described at the modeling level. Applications at level IV can be made available as bytecode on globally distributed application servers. When opening a folder to process an event, the workflow system generates a web page corresponding to the respective processing step. This web page starts the applet and calls the application server. Once connected to the applet, users execute the functions, after which the data is returned and passed on to the workflow system.

Data also includes information on the time and duration of the process in which the workflow system is aggregated and returned for process control. The process supports any operating system or hardware platform. The applets required for processing, including the object to be processed, are uploaded via the browser's front-end. Users can execute functions decentrally. Clients have immediate access to every method. After processing, transformed objects are returned to the server, passing them on for further processing.

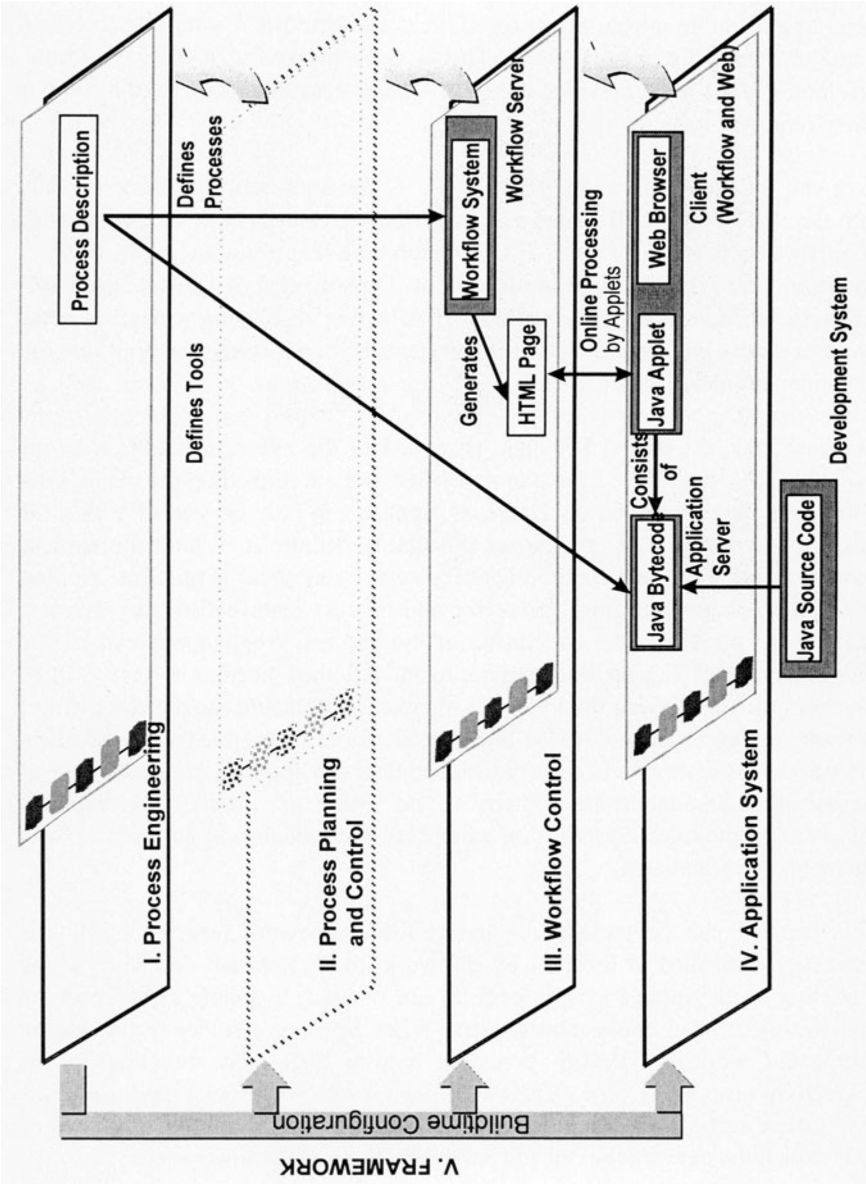


Fig.. 54 Linking applets with ARIS house of business engineering

While events are being processed, workflow servers are in constant control of any servers being started or data processed. Supervisors are able to query information on time and duration of the process, and can query process states at any given time, although it is not possible to access objects while being processed *(see Binas-Holz, Java Programmierbuch 1996, particularly pp. 98; Goldammer, HTML-Script ruft Java Applet 1996; van Hoff/Shaio/Starbuck, Java-Applets 1996).*

D.IV.2.4 Standardization Efforts

In order to assemble application systems from heterogeneous modules, standardized interfaces are necessary, particularly for communicating with client server environments. Various organizations such as the object management group (OMG), the open application group (OAG), as well as industry standards of key vendors (Microsoft, SAP, et al.) are attempting this standardization.

OMG is a cross-vendor organization. In early 1998, it had more than 800 members. Its object management architecture (OMA) provides a framework for distributing and enabling the interaction of object oriented software modules in networked, heterogeneous systems (see Fig. 55).

The centerpiece of this architecture is the object request broker (ORB), responsible for the interchange of messages between objects in heterogeneous platforms in accordance with the common object request broker architecture (CORBA) standard.

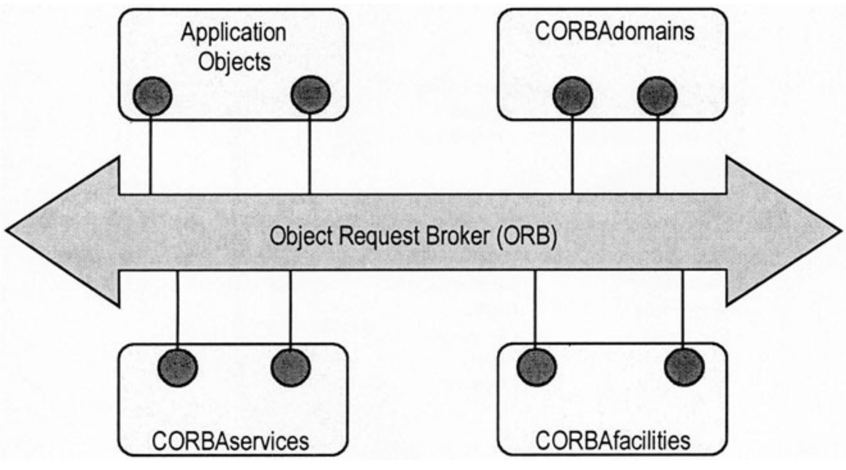


Fig. 55 Object management architecture
(from OMG, Common Business Objects 1996, pp. 3)

CORBA is a standard enabling the interaction of client objects and server objects. A client object can start a method using ORB, without having to „know“ the location of the server object.

OMG’s tasks involve describing the CORBA standard, among other things. The actual implementation process is left up to various vendors. One available commercial implementation is ORBIX by IONA Technologies, Ltd.

Standardization of the application objects is not taken any further. However, efforts are under way to define so-called „common business objects“ (CBO; see below). CORBAservices provide basic services such as consistency reviews.
CORBAfacilities, comparable to an all-purpose class library, provide standard functions prevalent in many applications, reducing application efforts of the user. CORBAdomains offer industry-specific functions such as for CAD systems.

Microsoft has also developed standards for object oriented component linking. COM (component object model) for single user systems and DCOM (distributed COM) for distributed systems are concepts for establishing communication between objects. Interfaces with the CORBA standard are currently being developed.

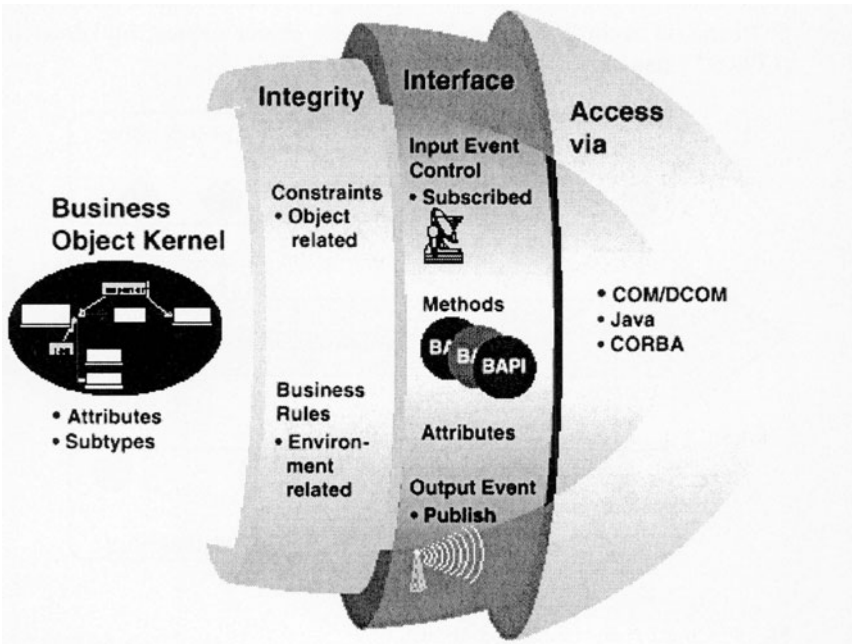


Fig. 56 SAP business object
(from SAP, White Paper Business Objects 1997, p. 7)

SAP has also developed a concept for communicating with business objects, called BAPI (business application programming interface). BAPIs are implemented as SAP business object methods, enabling open access to business access functions. Fig. 56 illustrates the structure of SAP business objects.

An SAP business object constists of a business object kernel which contains the core business logic. The second layer contains business rules responsible for integrity. The third layer contains the object methods, attributes as well as the input event control, output events and BAPI definitions. Supported interface standards include COM/DCOM, CORBA, and Java.

In an enhancement of this concept, business objects are grouped into packages, so-called business components. In its first stage, the entire R/3 system is classified in three components (HR, LO and FI/CO). Approximately 10 newly developed components have been included (e.g., Business Engineer and Business Information Warehouse). Communication between these components is enabled by the ALE concept (application link enabling), comparable to the communication between independent R/3 systems and between R/3 and R/2 systems (see Fig. 57). The ALE concept is also supported by the open application group (OAG), ensuring further development of a general standard. It also enables communication with Internet applications.

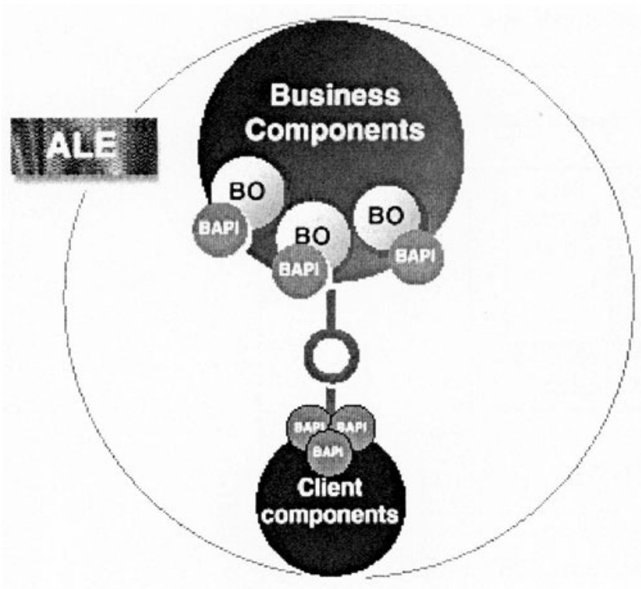


Fig. 57 Business components
(from SAP, White Paper Business Frameworks 1996, p. 6)

BAPI and ALE messages are methods for accessing business objects or components. The main difference is their granularity. Whereas an ALE message is capable of triggering a complex application via the Internet, such as an entire order processing procedure, BAPIs use more detailed methods, say, availability reviews *(see Zencke, BAPI 1997).* ALE messages can be processed asynchronously, while BAPIs run synchronously.

OMG has also championed the standardization of business objects, based on CORBA results. Its „request for proposal" (RPF-4), published in 1996, challenged the industry to make suggestions regarding the standardization of CBO (common business objects), the purpose being to define basic applications from which specific objects can be constructed. Fig. 58 illustrates how CBOs are embedded. For example, a CBO can be an object used for sales processing or financial accounting. Some suggestions addressed the issue of calculating currency exchange rates (brought forward by IBM) or referred to workflow applications. In order to describe objects, certain guidelines must be met *(see OMG, Common Business Objects 1996, pp. 17),* regarding:

* CBO designation,
* Attributes of the objects,
* Relationships between the CBOs,
* Conditions regarding CBOs and their relationships,
* Rules of the business process regarding CBOs and their relationships,
* CBO interfaces and methods, specified in OMG IDL.

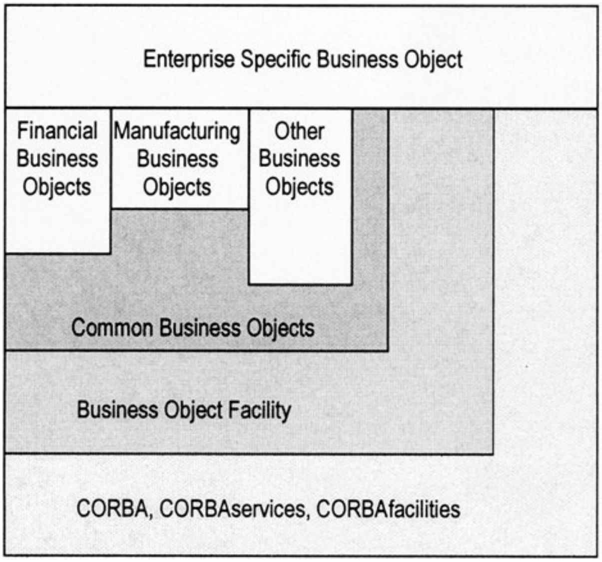


Fig. 58 Embedding common business objects
(from OMG, Common Business Objects 1996, p. 21)

These activities make „assembly line“ software development more realistic then ever. One of the key contributors to this approach is the framework concept.

D.V Frameworks

D.V.1 The Framework Concept

Although the object oriented approach has enhanced the module concept with its inheritance principle, making modifications easier without creating new module varieties, the framework concept goes yet a step further. It features various components and integrates them into a particular application (see Fig. 59). At first glance, this sounds like a business object. However, it also includes infrastructure components such as workflow systems, modeling tools and middleware, linking the business objects into an application within the framework. What's more, the object oriented inheritance principle is replaced by the composition principle.

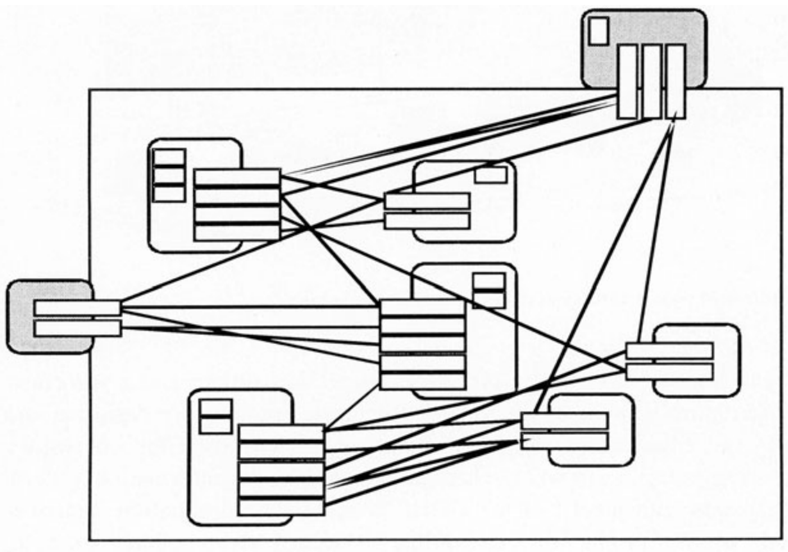


Fig. 59 Framework with exchangeable components
(from Pree, Komponentenbasierte Softwareentwicklung 1997, p. 7)

Components in a framework are interchangeable („hot spots“) when users can interchange them with their own components, in accordance with their own requirements. In Fig. 59, these components are shaded. Components that cannot be interchanged are conversely called „frozen spots“. Adapting frameworks by interchanging components is known as „composition“ and is an alternative to

object oriented adaptation where subclasses are characterized by inheritance. Thus, frameworks are incomplete application systems, that may be customized by the user switching components. Therefore, this approach makes both, components and the architectural know-how for linking the components, reusable.

Framework products focus on the frameworks themselves and on the inclusion of the middleware -- bridging the gap between application software, operating system and hardware.

We have tried to point out the analogy between industrial production systems and information systems. This analogy would seem particularly obvious when information systems are driven by workflows and frameworks.

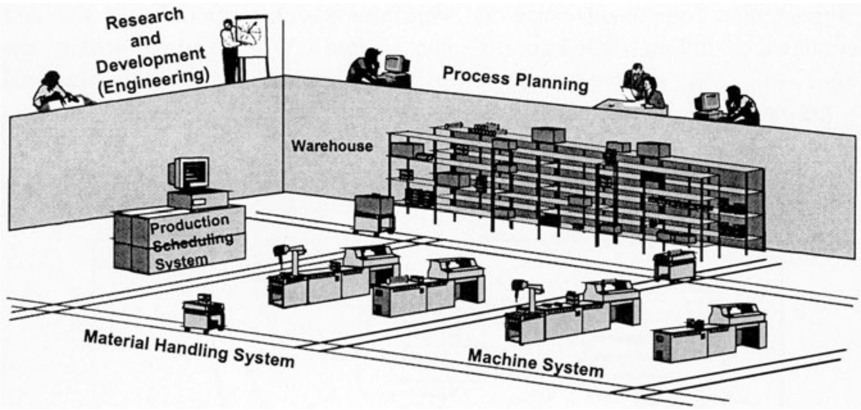


Fig. 60 Industrial production system

Fig. 60 and Fig. 61 illustrate an industrial production system and a workflow driven information system in equivalent structures. Products are designed and described in engineering. Process planning determines the necessary manufacturing processes in work schedules. Regarding the information system, this corresponds with level I of the HOBE concept. The production system is controlled by a production scheduling system, also called leitstand, corresponding with level II of the HOBE concept.

The material handling system bridges the gap between the warehouse, where the objects to be processed are stored, and the machine system executing the processing functions. The process is executed in accordance with the work schedule. The material handling system corresponds with the workflow system of level III in the HOBE concept. Storage and function execution of level IV correspond with the database and the business objects.

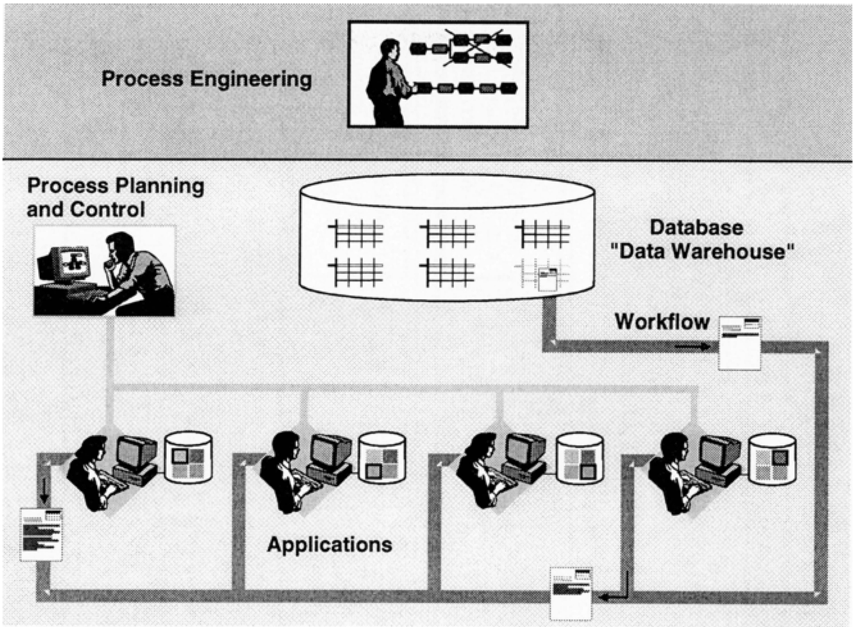


Fig. 61 Workflow driven information system

We hope to have made the analogies between these illustrations understandable. In information systems, the HOBE concept classifies the entire system in the warehouse (data storage), material handling system (workflow system) and function execution (business objects). Levels I and II are represented by product and process modeling -- and process planning and control -- respectively. Structuring information systems into subsystems streamlines their development, control and makes customizing more flexible.

D.V.2 Realization Concepts

Today, realization concepts using the term „framework" are in various stages of completion. Nonetheless, they illustrate the future importance of this concept.

D.V.2.1 ARIS-Framework

ARIS framework, developed as a prototype by IDS, is compatible with the HOBE concept -- with ARIS Toolset providing modeling and analysis tools at the process engineering level.

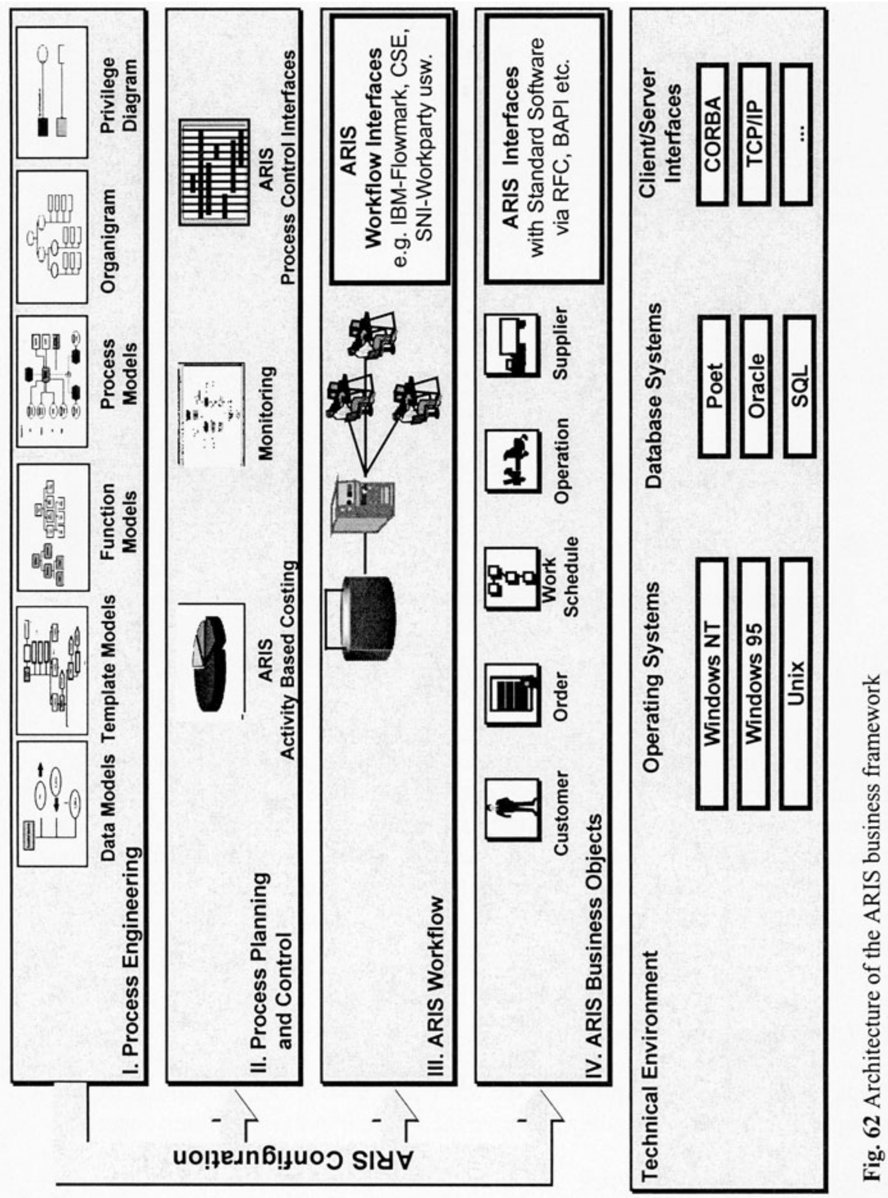


Fig. 62 Architecture of the ARIS business framework

Fig. 62 Architecture of the ARIS business framework

In business process planning and control, there are special integrated products for activity based costing, monitoring, scheduling and capacity control as well as interfaces with third-party products such as MS Project.

At the workflow level, ARIS workflow provides prototyping functionality and interfaces with approximately ten different workflow systems.

At the application level, generic business objects for logistics solutions are available, in addition to interfaces for starting standard software solutions such as SAP R/3 via RFCs and, subject to availability, BAPI.

Application know-how is stored in reference models, populating other levels with content, at level I. By customizing reference models, generic business objects developed at level IV can be adapted to specific applications. Fig. 62 illustrates the architecture of the ARIS framework, with business objects administrated on the application server by an object manager. An independent interface with relational databases allocates the data to the application server. A CORBA gateway provides an interface with external systems which can be started by events. ARIS workflow controls the processes within and amongst the business objects, and with external systems as well.

Window clients are linked with the application server by CORBA interfaces.

Business objects can be configured at the modeling level, using model types at level I.

Attributes are added or suppressed by using data models. Business object screens are developed by means of template models. Business object functions can be selected using function models. Process models configure the allocation between functions and models. They also control processes within the business objects. Function and data privileges are defined by org charts and privilege diagrams. Modeling methods and configuration options are discussed in detail in *Scheer, ARIS - Business Process Modeling 1998.*

D.V.2.2 SAP-Framework

SAP business objects and components shown in Fig. 56 and 57, respectively, illustrate key points of the SAP framework concept, consisting of the SAP reference model where business processes are defined by EPCs, ALE integration technology, SAP business workflow, BAPI interface technology for business objects and business objects themselves *(see SAP, White Paper Business Framework 1996, p. 15).*

Although not mentioned in the above list, we should also include the SAP Business Engineer (BE), along with its configuration options, in the framework concept. A BE overview is given in Fig. 63 *(see SAP, White Paper Business Framework 1996, p. 9; Schröder, Business Engineer 1997).* Despite individual development of various components (BE, workflow, data modeling, etc.), solution integration is becoming increasingly popular. The SAP framework not only integrates SAP software solutions, but third-party solutions as well. This is the key for providing comprehensive vertical market solutions. At the same time, users can use the SAP framework and technology as a toolbox for developing their own SAP R/3 enhancements.

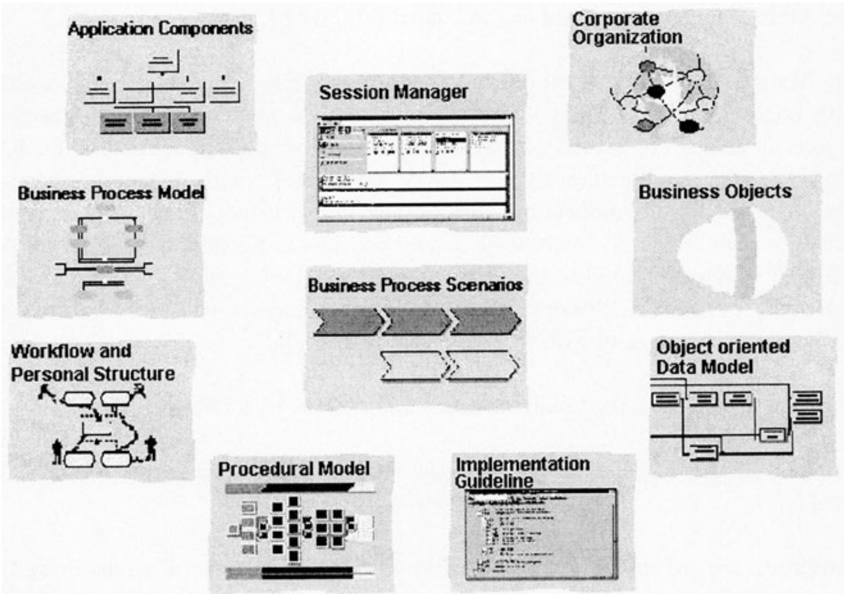


Fig. 63 SAP Business Engineer components
(from SAP, White Paper Business Framework 1996)

D.V.2.3 SNI-ComUnity

Siemens-Nixdorf provides a framework consisting of the Workparty workflow system, the ComUnity configuration tool and various business objects for industrial applications. Linking ARIS modeling products of levels I, and (at a later point in time) of level II, SNI creates a conceptual design more or less in accordance with the HOBE concept. Workparty's technical platform architecture is shown in Fig. 64, characterized by frequent use of Microsoft standards.

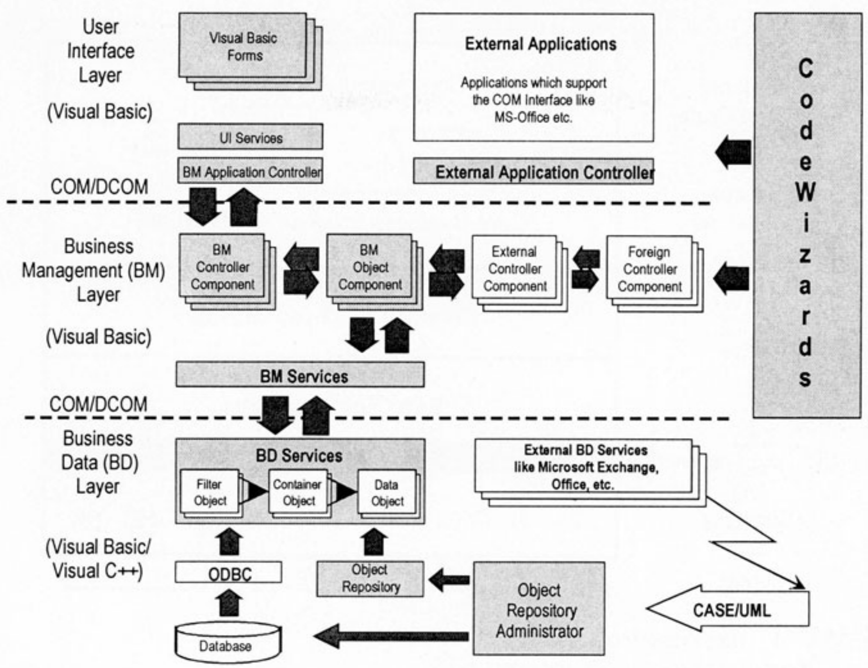


Fig. 64 SNI-framework ComUnity architecture
(from Siemens-Nixdorf, ComUnity 1997)

D.V.2.4 IBM's San Francisco Project

With its San Francisco project, IBM aims to enable faster and less expensive application development for SME software vendors. The San Francisco project consists of various layers, as shown in Fig. 65. The application designer at the top accesses the other layers. The Java virtual machine makes application solutions independent of system and hardware platforms.
The base enabling layer is divided into three categories: kernel services; base object model classes; utilities.

- Kernel services are based on the object services defined by OMG. They are written in Java and are partially adapted to Java functionality and specific IBM requirements.
- Base object model classes contain mechanisms for storing or identifying objects.
- Featuring basic graphical user interface (GUI) components and session management functionality, utilities provide features for controlling contentions. Although parts of this functionality are already provided by the operating system, these features ensure integration and a common „look and feel".

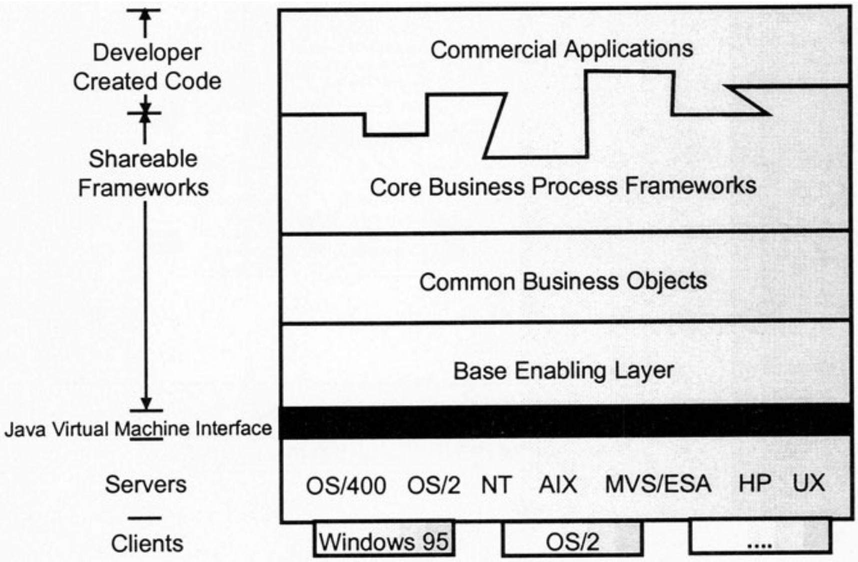


Fig. 65 IBM -- San Francisco project
(from IBM Shareable Frameworks 1996)

The Common Business Objects (CBO) are applicable to many vertical markets. Software developers need to complete individual functions or add specific characteristics of the particular company. This might include address management, payment terms, scheduling, or data interchange functions such as EDI. These CBOs are comparable to OMG CBOs which would explain IBM's interest in having CBOs certified by OMG. Core business process frameworks provide objects for order management, warehouse management and corporate accounting. According to IBM's estimates, CBOs and core business process frameworks make up about 40% of any given application. The balance consists of screens, country- and industry-specific features, business rules and additional individual application functions.

D.V.3 Effects on the Software Industry

Software development will change thanks to the opportunities provided by componentware and frameworks. Just as in the hardware industry, where vendors no longer manufacture entire systems vertically, but rather have moved on towards horizontal manufacturing of processors, peripherals, operating systems, etc., which are then assembled by the vendor, this will also occur in application software development. For hardware, the prerequisite was the development of standards for processors, operating systems, databases and communication networks.

Moreover, application software will also need standards for components and frameworks. These standards will make it possible to adapt components from individual manufacturers to various frameworks.

We believe that the software industry will reduce its depth of manufacturing. The market will consist of vendors specializing in entire solutions and assembling components in (their) frameworks, -- and component providers. Between these two extremes, vendors of subsystems will provide intermediate products.

Components will be selected and replaced in accordance with „best of breeds" principles, focusing on business know-how and modeling, in accordance with the assembly. Thus, the engineering level of the HOBE concept with its modeling methods, and integrating and customizing links with workflow and business objects -- will become focal points of software development.

Just as in the automotive industry where car manufacturers provide development (i.e., engineering), logistics and assembly expertise, vendors of complete solutions will need to provide the content and modeling techniques featured at level I, in addition to mastering frameworks. Furthermore, they will have to keep an eye on and evaluate component vendors, and also establish relationships between the manufacturers and suppliers. Vendors of complete solutions will also have to provide worldwide consulting and service expertise, i.e., features only large corporations are capable of offering. This expertise will be necessary to guarantee the integrity of their total solutions. Smaller software vendors will concentrate on developing individual components and subsystems.

On the other hand, we should not overestimate the opportunities of a componentware market, from which large solution vendors would be able to pick and choose. In fact, the metaphor of Lego blocks that can be assembled freely, is not applicable. Lego blocks are identical and can be assembled according to a blue print. It is not necessary to understand the internal structure of the blocks. All one needs to know is that their „links" and their sizes are standardized.

Software components, however, contain application information -- different for each component. In order to assemble them, we need to understand their technical interfaces and their application logic in-depth. This determines whether a certain component can be assembled or not. Comprehensive documentation of the components is a main prerequisite for the correct assembly of a component. Solution and component vendors should collaborate on complex modules, just like industrial suppliers in the aerospace or automotive industries. In these industries, exchanging product information and close cooperation during development is called simultaneous engineering *(see*

Scheer, Business Process Engineering 1994). A lot of the expertise and multiple procedures of simultaneous engineering are applicable to the software industry. Unfortunately, in contrast to the manufacturing industry, it is more difficult for a software component vendor to protect confidential product know-how. Whereas the expertise of automotive suppliers in R&D and in manufacturing lies in know-how regarding procedures and machine resources, and whereas automotive suppliers need to continuously manufacture their products, software vendors only do have their development know-how. After close cooperation on a particular project, it would be very easy for a „partnering" solution provider to copy that know-how.

If the software industry is to focus on componentware full scale, this would stipulate effective protection mechanisms as well as a culture of trust and cooperation between component and solution providers.

E Modeling Standards in ARIS

Modeling in ARIS is the manipulation of elements, using ARIS views, phases, designations and methods -- for the purpose of describing business processes. Modeling is a creative process and can therefore not be completely directed by rules. However, if certain standards are observed, it is indeed possible to classify and understand third-party models, just as it is also a good idea to establish certain quality standards and observe them.

After discussing the basics of general modeling principles, we will focus on various issues pertaining to modeling levels (instance level, type level, meta level and meta^2 level), granularity, detailing and different model variants.

E.I Generally Accepted Modeling Principles

The term "generally accepted modeling principles" *(see Becker/ Rosemann/Schütte, Grundsätze ordnungsgemäßer Modellierung 1995; Galler, Vom Geschäftsprozeßmodell zum Workflow-Modell 1997, p. 124; Reiter/Wilhelm/Geib, Multiperspektivische Informationsmodellierung 1997)* is borrowed from the term "generally accepted accounting principles", GAAP. In modeling, this requires as many degrees of freedom as possible and, at the same time, aiming to make quality control easier to understand *(see Maier, Qualität von Datenmodellen 1996).*

The following rules, continuously enhanced in a research project of the German government *(Projekt „GoM", Fördergebiet Softwaretechnologie; Az.: 523-4001-01 IS 604 A)* are integrated with the ARIS concept and the supporting ARIS tools.

* **Principle of correctness:** The correctness of models depends on correct semantics and syntax, i.e., whether syntax of the respective meta model is complete and consistent. "Semantic correctness of a model is measured by how closely it complies with the structure and behavior of the respective object system" *(see Rosemann, Komplexitätsmanagement in Prozeßmodellen 1996, p. 94).* In real-world applications, compliance with these requirements can be proven only after simulation studies have been carried out or other similar efforts have been made. The ARIS Toolset provides a simulation tool that is well suited for this purpose. This tool features a wide range of rules for

reviewing model syntax, not only ensuring that every function is triggered by an event, but also leads to an event, etc.

* **Principle of relevance:** Excerpts of the real-world object system should only be modeled when they correspond with the purpose of the model. Models should not contain more information than necessary, thus keeping the cost vs. benefit ratio down to an acceptable level.
* **Principle of cost vs. benefit:** One of the key factors ensuring a good cost vs. benefit ratio is the amount of effort necessary to create the model, the usefulness of modeling the scenario and how long the model will be used.
* **Principle of clarity:** "Clarity" ensures that a model is understandable and usable for its users. It also determines how pragmatic the relationship between the model and the user is *(see Rosemann, Komplexitätsmanagement in Prozeßmodellen 1996, p. 99).* Because models contain a large amount of information regarding technical and organizational issues, only specialists are usually able to understand them quickly. Once models are broken down into sub-views, individual views are easier to comprehend.
* **Principle of comparability:** Models created in accordance with a consistent conceptual framework and modeling language are comparable if the objects have been named conforming with established conventions and if identical modeling objects as well as equivalent degrees of detailing have been used *(see Rosemann, Komplexitätsmanagement in Prozeßmodellen 1996, p. 102).* In models created with different modeling languages, it is important to make sure that their meta models can be compared.
* **Principle of Systematic Structure:** This principle stipulates that it should be possible to integrate models developed in various views. This requires a single meta model across various views, something of which the ARIS information model is capable.

E.II Modeling Levels

The ARIS concept was designed at the application independent meta level (see Fig. 12). Due to the fact that terms permissible at this level are also valid for underlying application types and instances, the ARIS concept is automatically applied to underlying modeling levels as well.

Fig. 66 shows an example of ARIS views at the requirements definition level. For every view, a typical designation -- showing the class-element relationship in accordance with the designation of the underlying abstraction level -- is depicted.

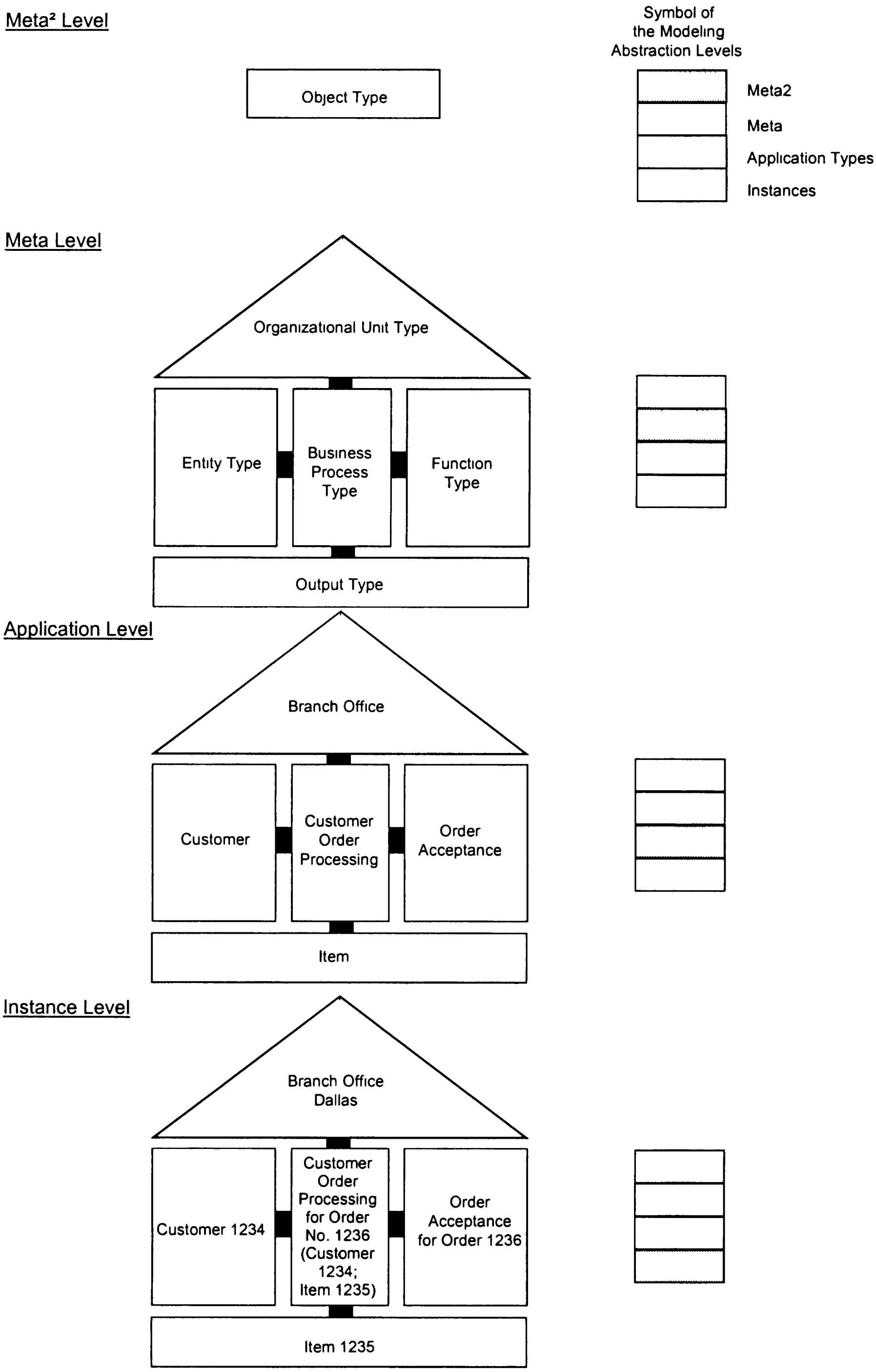


Fig. 66 ARIS modeling levels

At the meta² level, only the general designation "object type" is defined. In this class, designations of the meta level are stored as elements. The ARIS information model is developed at the meta level, i.e., this is where general designation classes describing the business process, along with their relationships, are defined. Real-world applications are modeled at the application level with business information systems generally being developed at the description level, giving these modeling methods special significance. Individual instances are modeled at the instance level. These various processes are executed at run-time.

Each abstraction level is displayed by the appropriate symbol (see Fig. 66, right hand side of diagram).

Models are usually created at the type level, during business process build-time. This makes the models "invulnerable" to any instance modifications, such as when new items are added to the data model or when individual employees are removed from the organization model. However, if the same instances are always used in a certain business process type, they can also be used for build-time modeling. This would be the case when functions can only be carried out by a certain specialist or if the same data instance is used over and over again or if a model always refers to a certain organizational unit and not to its type. This is the reason some models combine instance levels and type levels.

In order to better understand description levels, a draft of the particular model administration standard, based on the ARIS Toolset, is made.

All ARIS Toolset models are stored and administered at the meta² level. This makes them independent of methods because all the method and view specific designations are instances of this general object type. Thus, it is possible to incorporate new modeling methods in ARIS Toolset, (generally) without any program modifications.

At the meta² level, the class OBJECT TYPE uses meta level modeling objects as instances (e.g., function types, organization unit types, output types, etc.) (see Fig. 67). The same operations (create, delete, display, drag and apply hierarchies) can be applied to every modeling object, which defines them in the class OBJECT TYPE. In Fig. 68, the class OBJECT TYPE is shown as a table. When new modeling objects are introduced by new methods, they are entered as new instances of this class, i.e., as lines in the object type table.

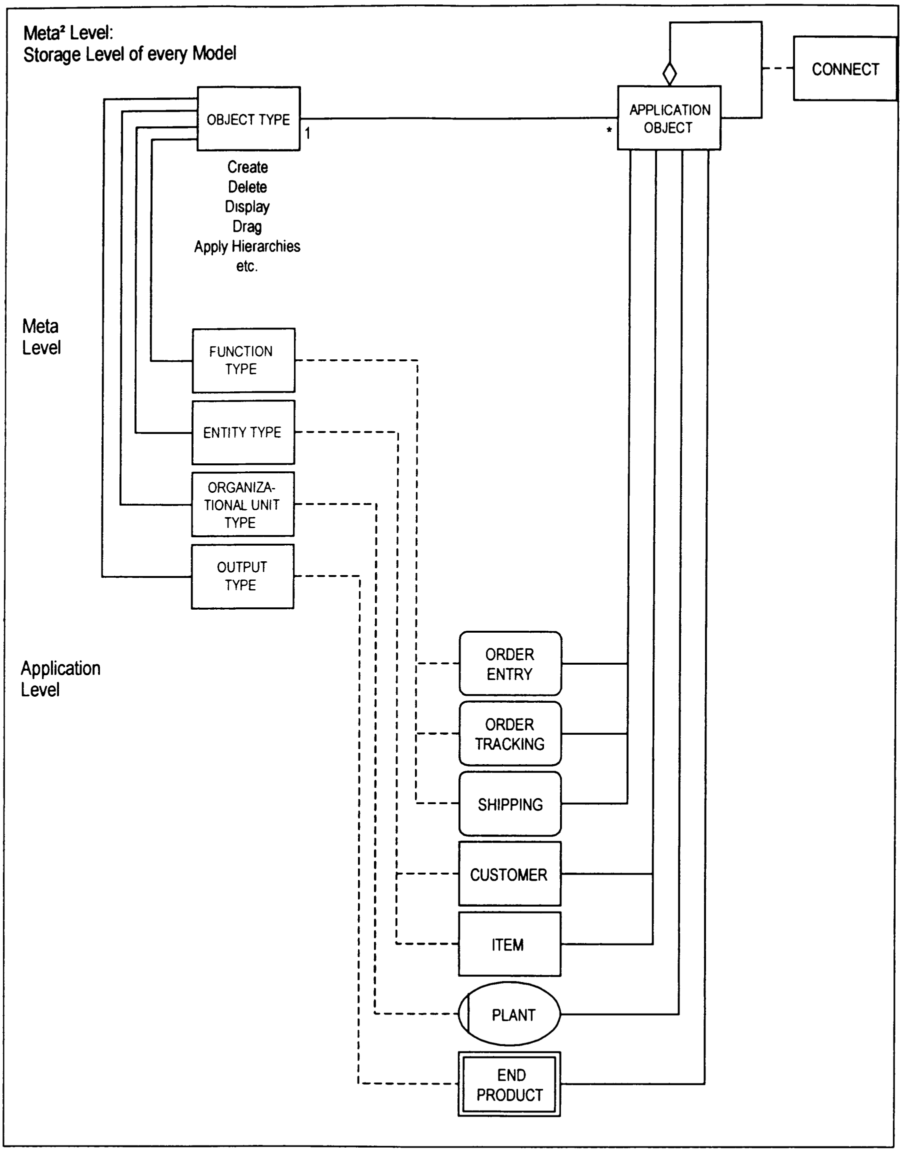


Fig. 67 The model administration standard in ARIS toolset

Object Type	Object Type Number	Designation	Illustration Type
	1	Function Type	Rounded Rectangle
	2	Entity Type	Single Line Rectangle
	3	Organizational Unit Type	Oval
	4	Output Type	Double Line Rectangle

Fig. 68 Object type table

Application Object	Application Object Number	Object Type Number	Designation
	1	1	Order Entry
	2	1	Order Tracking
	3	1	Shipping
	4	2	Customer
	5	2	Item
	6	3	Plant
	7	4	End Product

Fig. 69 Application object table

Object Type	Object Type Number	Designation	Illustration Type
	:		
	4	Output	Double Line Rectangle
	:		
	10	Function Instance	Rounded Rectangle
	11	Entity Instance	Single Line Rectangle
	12	Organization Instance	Oval
	13	Output Instance	Double Line Rectangle

Fig. 70 Object type table enhanced by instance types

Application Object	Application Object Number	Object Type Number	Designation
	7	4	End Product
	:		
	10	4	End Product 1234
	11	4	End Product 1235
	12	2	Customer M
	13	2	Customer N

Fig. 71 Application object table enhanced by instance types

Models at the application level are the instances of modeling objects at the meta level. In Fig. 67, this is suggested by the dashed lines. For storage purposes, the class APPLICATION OBJECT is introduced at the meta² level. This class contains all the instances of the meta level objects and is therefore connected with the class OBJECT TYPE by the *:1 association. Storage relationships are illustrated by solid lines. Model lines between the modeling objects are entered by the link CONNECT between APPLICATION OBJECTS. This context is shown in Fig. 69. Whereas the class names of the meta level are elements of the class OBJECT TYPE, the application objects are elements of the meta classes.

When instance models as well are maintained by ARIS (such as for workflow applications), the interpretations of the ARIS meta and meta² models are enhanced. The structure of the meta² model in Fig. 67 does not change, whereas the permissible instances do. The meta² model in the class OBJECT TYPE (see Fig. 70) now adapts the various instance descriptions of the application objects as well as the real-world instances of these application objects (see Fig. 71). Thus, application types and their instances are entered in these tables. This also holds true for the association CONNECT and model lines.

Thus, every model in ARIS Toolset is logically stored in a few large tables. These tables are implemented in an object oriented database (POET), making more detailed access structures possible (such as for accessing comprehensive models) for improved performance.

E.III Degrees of Granularity and Detailing

Models can be created using designations of varying granularity, such as depicted in the example in Fig. 72, showing the requirements definition of data classes and links in an application area. Initially, the class diagram is aggregated into cluster designations and then into the sales model. There are 1:* - "part of" associations among each of the three designation levels.

Boxes in three-tier pyramids indicate the granularity of a model (see Fig. 72).

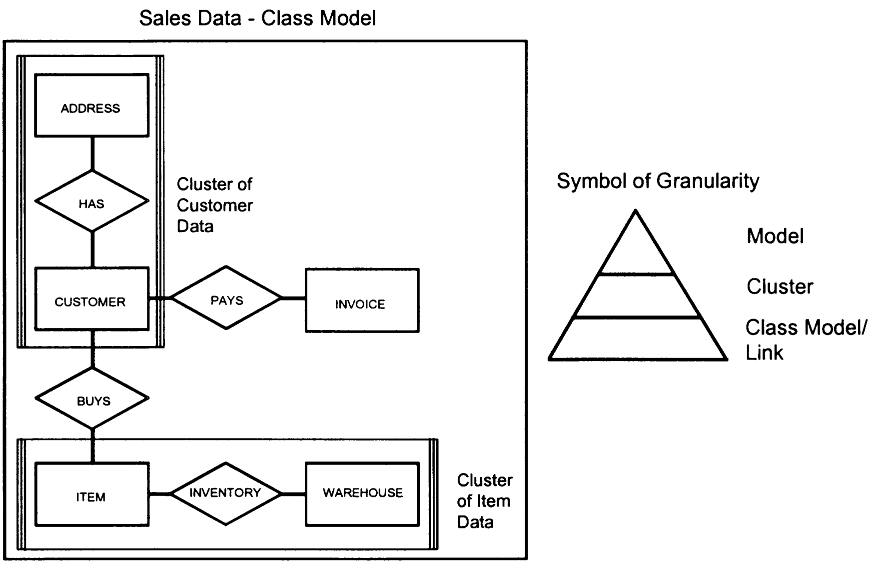


Fig. 72 Granularity of a model

Hierarchies must definitely be applied when describing large application areas. Fig. 73 shows an example of how the ARIS Toolset applies hierarchies to models, taken from the IDS Prof. Scheer GmbH reference model for sales logistics.

Fig. 74 shows the main granularity steps of SAP R/3's Business Engineer BE module. Large function areas (sales, production, etc.) of a particular vertical market are combined at the level of a business scenario; business processes address various process alternatives of a function area. Within a process alternative, various function alternatives (business functions in a business process) are illustrated.

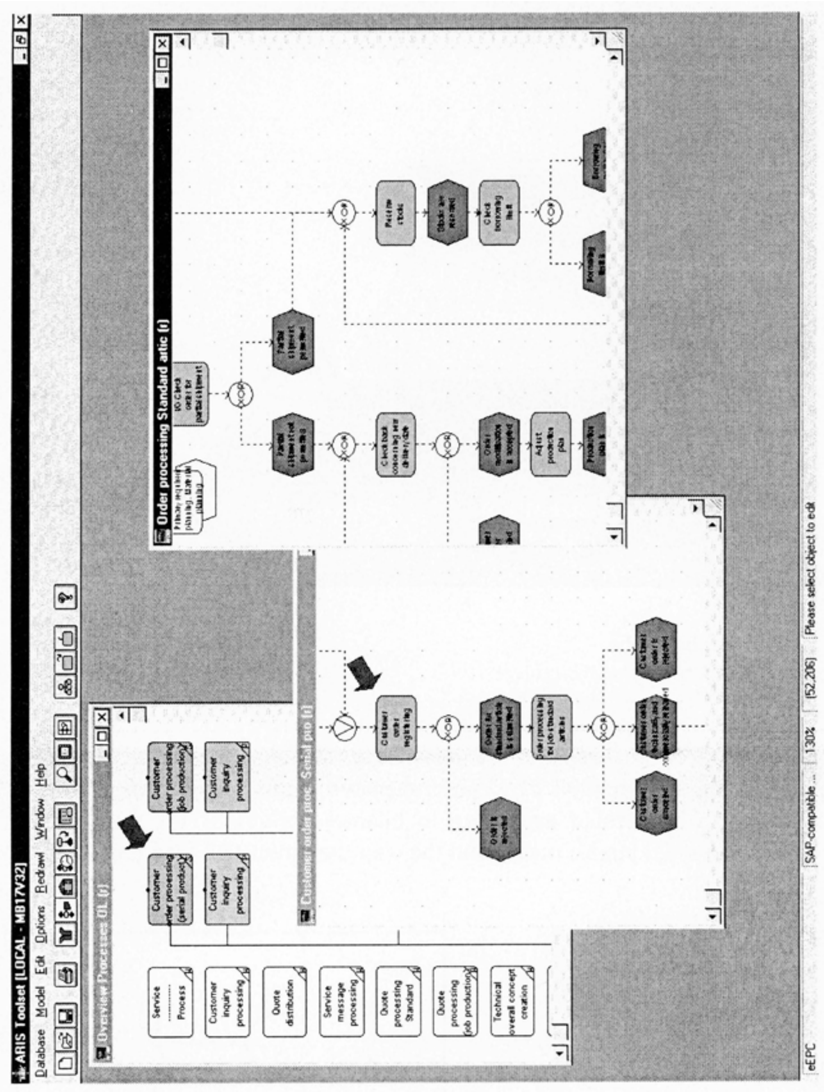


Fig. 73 Example from ARIS, depicting different levels of a reference model

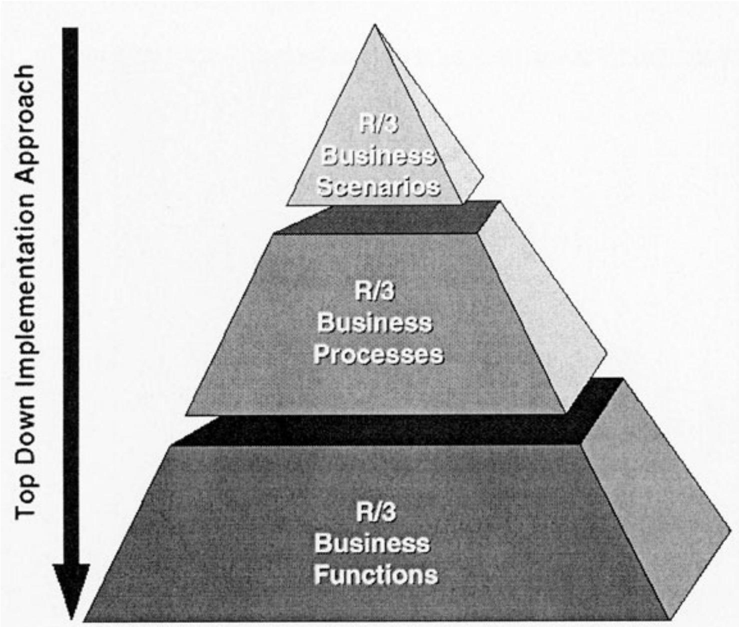


Fig. 74 SAP model hierarchy
(see Schröder, Business Engineer 1997)

Furthermore, models are capable of describing various sub-areas of entire complexes. Whether sub-models of a comprehensive model, or enterprise models – all of them can be created according to business criteria. This fact can be visualized by individual puzzle pieces and the way they interlock (see Fig. 75).

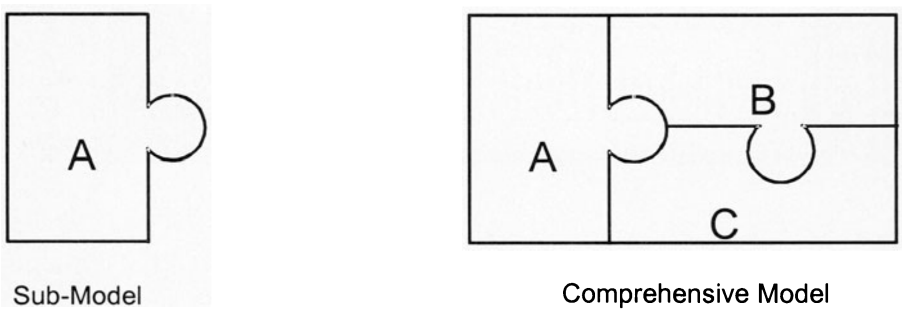


Fig. 75 Symbols of a sub-model and a comprehensive model

E.IV Model Variants

While discussing CPI and model versions, we also reviewed model variant administration, with model versions created over time and differentiated by time stamps. Occasionally, multiple concurrent model variants – each one created for a different condition of the same application case -- are also possible. There are two basic methods for creating variants:

* Selecting a variant from an existing overlying model. Here, knowledge regarding the relationship of the objects is passed on from the overlying model, in addition to the objects themselves.

* Constructing a variant from generic building blocks.

In the CIM Analyzer developed in 1991 at the IWi, functions necessary in enterprises were selected in accordance with rule knowledge from a comprehensive function tree and business attributes *(see Jost, EDV-gestützte CIM-Rahmenplanung 1993, pp. 33).*

SAP's R/3 system utilizes an enhanced approach. At the industry scenario level (see Fig. 74), function areas are determined on a QA basis (on-line question and answers). Then, in similar fashion, an alternative process is established by redlining the unnecessary objects in an overlying process model. According to this same schema, parameter values within the process are then determined for alternative functions. The basic concepts of this procedure were included in Nixdorf's COMET system *(see Scheer, Efficient Information Management 1991).*

The CIM Analyzer concept and SAP's BE take rule knowledge into account. The CIM Analyzer concept captures the context between a business property (such as serial manufacturing) and the necessary functions (such as inventory). In BE, integrity conditions are defined in accordance with rule knowledge and links between functions (for example, if function A is obsolete, function C will be too) can be used to streamline on-line selection.

The author Remme suggests a construction-oriented approach for creating variants *(see Remme, Konstruktion von Geschäftsprozessen 1997; Remme, Organisationsplanung 1997; see also Lang, Gestaltung von Geschäftsprozessen 1997),* which is integrated as a prototype with ARIS Toolset. Remme's theory states that enterprises are the result of engineering decisions (see Fig. 76).

Engineering Decision	Organizational Effects
Switching production from customer-specific products to standard products with variants	For example, uncluttering engineering and customer order processing procedures; developing standard work schedules; implementing a variant administration system
Return of end products after product life	For example, implementing engineering that is easy to disassemble; delivery of waste disposal capacities

Fig. 76 Examples for engineering decisions and their effects
(from Remme, Konstruktion von Geschäftsprozessen 1997, p. 90)

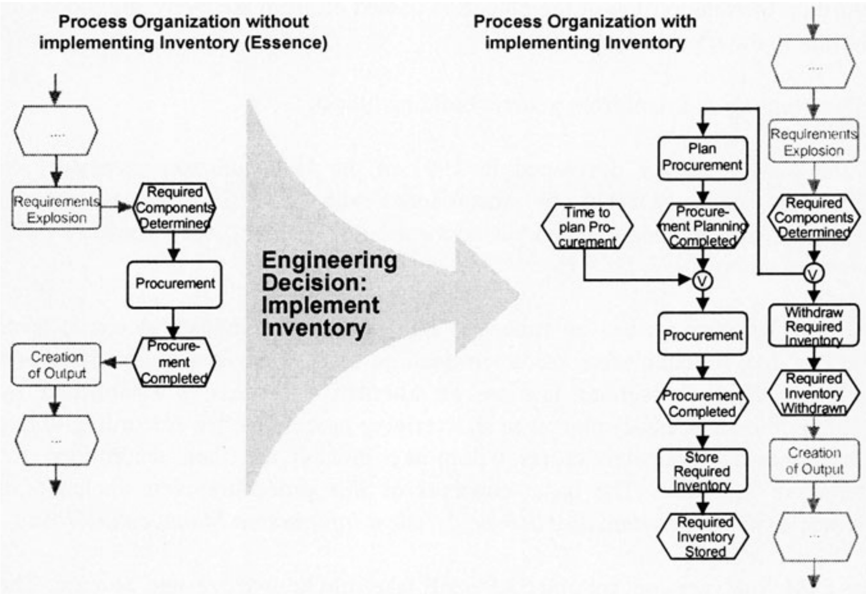


Fig. 77 Effects of engineering decisions
(from Remme, Organisationsplanung 1997, p. 13)

Enterprises that only fulfill basic functionality and are not influenced by any engineering decision are called "essences". Engineering decisions effect these essences. Organizational effects are defined as generic process particles, describing the process oriented effects for an enterprise when the enterprise makes an engineering decision. Continuously assembling the process particles in the initial situation leads to an application specific process model as depicted in Fig. 77, where the engineering decision "implement inventory" in the essence "procurement" is shown. As was the case when assembling a pre-fabricated assembly during production, the process particle is positioned in the essence and then connected with it.

Thus, ARIS models can be characterized by the ARIS views (5 views @ three phases each), modeling levels (meta^2, meta, application type, application instance), granularity, degree of detailing and variant designation.

F Comparing ARIS with Other Concepts

When the first edition of this book was published, only a few detailed architectural concepts for describing information systems were available, with which ARIS could be compared. In the mean time, theoretical discussion and real-world implementation of information system architecture has become more popular. Empirical information management studies show that the importance of IS architecture is acknowledged in real-world scenarios as well *(see Krcmar, Informationsmanagement 1997, p. 9; Nüttgens, Koordiniert-dezentrales Informationsmanagement 1995, p. 69).* Of the various methods for comparing architectures, the analysis of meta concepts now seems to be the most popular.

Although the key benefits of ARIS are its concept for integrating

* Architecture,
* Various methods and
* Tool support,

we will also compare approaches that focus on only one of these components. However, we would like to stress once again that ARIS' integrated approach offers special benefits in practical implementation. This is due to the fact that separate review of architectures and methods – without support of respective tools --, or support of graphics without a corresponding method concept (as with graphics tools), is not sufficient for real-world implementation.

The ARIS concept provides a reference framework for modeling methods, yet is independent of any particular method. Therefore, the methods used in ARIS are not irreversibly defined, but continue to be enhanced. A measure for the rich functionality of ARIS is not whether it supports a certain method, but also whether a new method can be logically classified in the ARIS framework and thus can be included in the range of methods. *Scheer, ARIS - Business Process Modeling 1998* discusses in detail various methods suitable for ARIS. The meta model presented in that work is a good starting point for comparing other methods.

We do not wish to compare tools at this time, but would instead refer to the publications listed below. Furthermore, various market analysts have published comparative studies *(see Long, Taxonomy of BPR Tools 1992; Finkeißen/Forschner/Häge, Werkzeuge zur Prozeßanalyse 1996).* The best-known study is probably that of the Gartner Group (see Fig. 78), which lists ARIS Toolset under the vendor's name, namely "IDS Prof. Scheer".

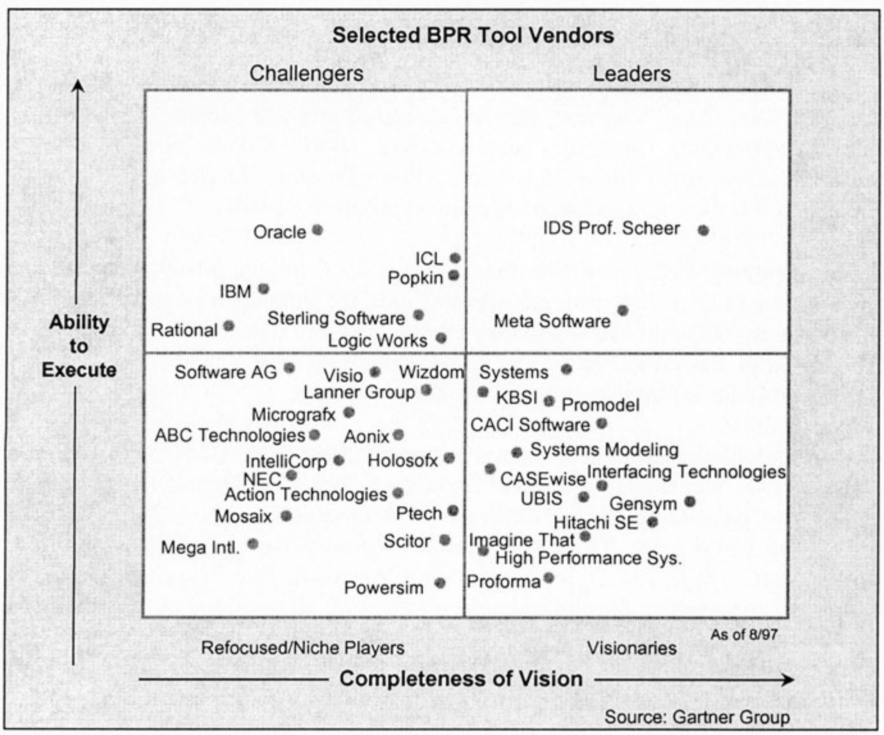


Fig. 78 Modeling tools
(Source: Gartner-Group 1997)

Next, let us compare other concepts with the ARIS concept.

F.I Object Oriented Modeling

Although object oriented concepts seldom use the term "architecture", due to their increasing importance we will first compare them with ARIS. Considering the properties of object oriented approaches, we will now take a closer look at their class creation, encapsulation methods and attributes as well as message exchange functionality.

The creation of object oriented models is based on their system theory goal of capturing, explaining and describing complex systems by using consistent standards. Systems are made up of a large number of components (sub-systems or elements), linked with one another by relationships. The purpose of modeling a system is to reduce the complexity of the object to be viewed by means of abstraction. In system theory, we can distinguish between system structure and system behavior. Therefore, object oriented modeling methods should be

distinguished by whether they aim to model only the structure -- or the behavior of the system as well.
In structure modeling, class creation is the first and foremost goal. Proponents of this concept are Coad, Yourdon, Rumbaugh and Shlaer and Mellor, among others *(see Coad/Yourdon, Object-Oriented Analysis 1991; Coad/Yourdon, Object-Oriented Design 1991; Rumbaugh et al., Object Oriented Modeling and Design 1991; Shlaer/Mellor, Object Oriented Systems Analysis 1988).*

Finding appropriate classes is thus the central task of these approaches. However, they unfortunately do not provide specific aids for creating classes. Therefore, these concepts frequently refer to knowledge attained in data modeling, especially in ERM. After the design stage, operations (methods) are linked to classes and dynamic behavior is supplemented by message exchange.

When modeling processes are aligned with system behavior, operations are the key issue. Proponents of this concept are Meyer, Wirfs-Brock and Jacobson *(see Meyer, Object-Oriented Software Construction 1988; Wirfs-Brock/Wilkerson/Wiener, Objektorientiertes Software-Design 1993; Jacobson, Object-Oriented Software Engineering 1996).* We would like to specifically point out the use case method developed by Jacobsen et al., highlighted by its use of the UML concept.

Despite the abstraction of irrelevant properties in the item to be viewed, a high degree of semantics is retained in object oriented modeling -- when a class is linked with its attributes, methods and associations, the purpose being to make it easier to intuitively understand a model. On the other hand, this also makes large models very complex and difficult to comprehend, even for skilled users. The goal of reducing complexity is to omit (abstract) unimportant elements and relationships in the system.

One of the main disadvantages of the object oriented approach is that it does not illustrate processes in a very detailed manner. Even with methods such as Use Case or with interaction diagrams, it is difficult to depict process branching, organizational aspects and output flows *(see Frank, Multiperspektivische Unternehmensmodellierung 1994, p. 136).*

State transition diagrams as well as actions and object flow diagrams (see Fig. 79, taken from the UML documentation) are process illustrations quite similar to EPCs. This figure shows the control flow between functions, the allocation of functions to organizational units and the flow of the processing objects, corresponding to the order in our example.

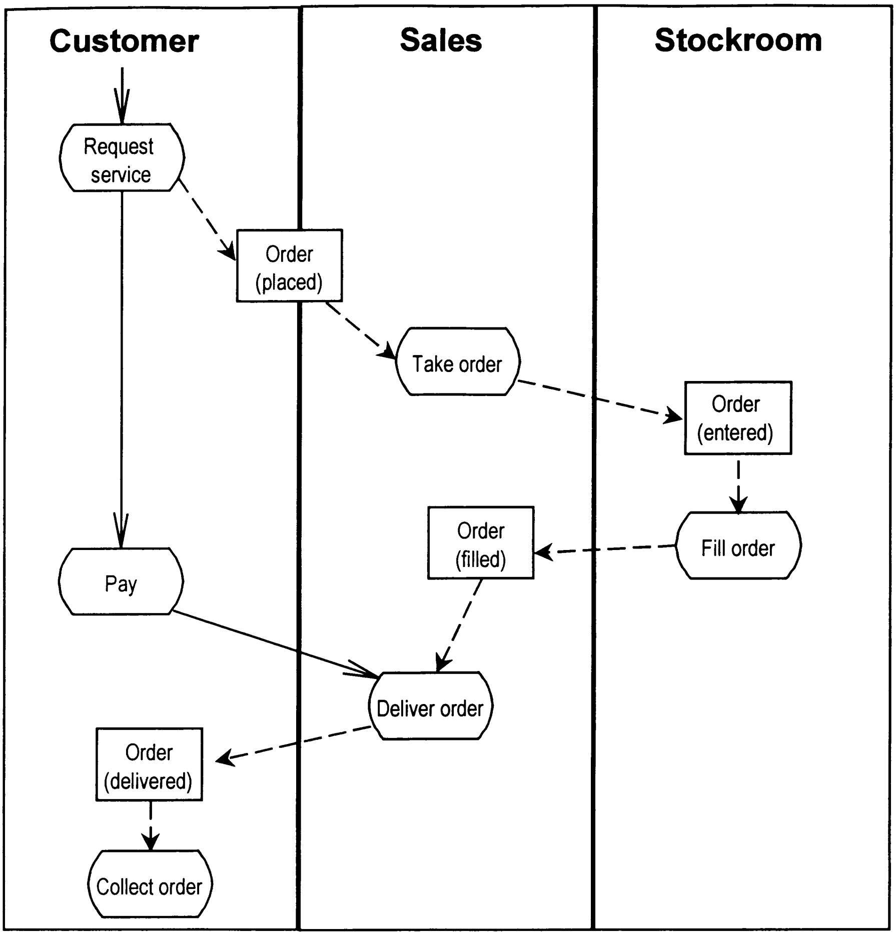


Fig. 79 Actions and object flow diagram
(from UML Notation Guide 1997, Fig. 56)

The order is an information object. On the other hand, it is also an output indicator, which is why the information service flow is also shown. In total, all the ARIS views are contained in the diagram type, although a framework for classifying the illustration elements and for enhanced illustration functionality within the individual views is missing. For additional comparison, the initial order processing example from Fig. 3 is illustrated in Fig. 80 in an actions and object flow diagram.

One advantage of object oriented modeling, however, is the close link of the models with implementation. This makes, for example, prototyping very easy.

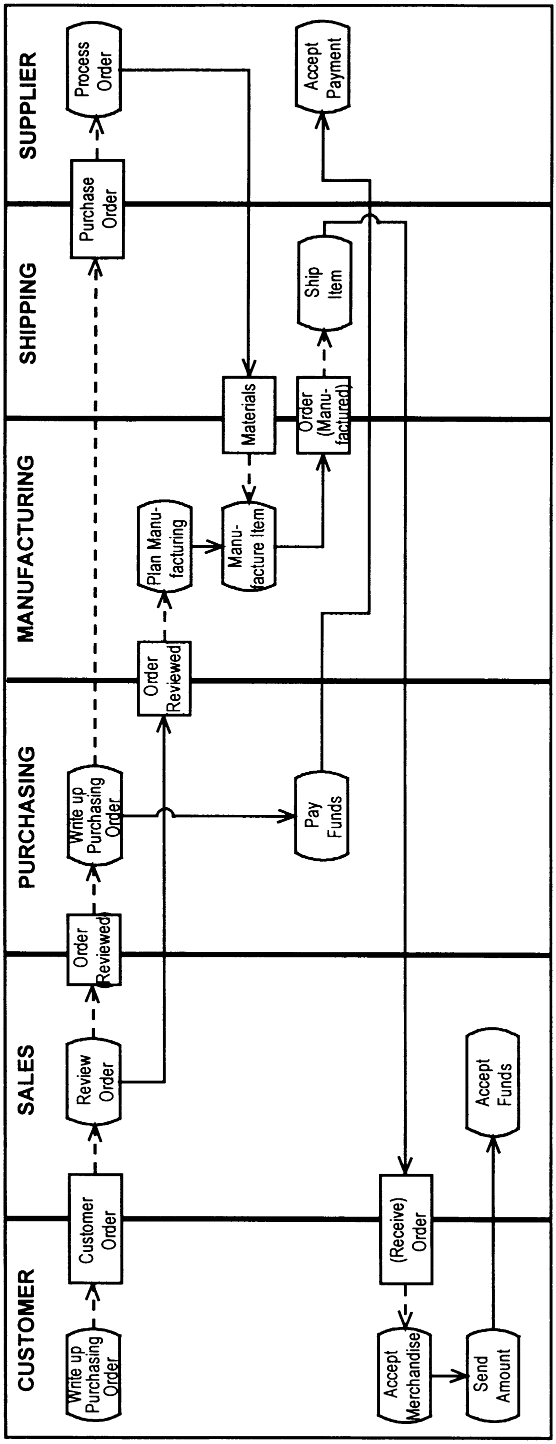


Fig. 80 Initial example from Fig. 3, illustrated in an actions and object flow diagram

ARIS' view concept reduces complexity by concentrating on one view dimension **and** abstracting. This also makes it possible to deploy very simple modeling methods such as organigrams, although ensuring cross-view consistency of the models is difficult. Control views or process views, where these model views are once again merged, are thus extremely important. They also classify the object oriented model. Therefore, view specific models can be regarded as an enhancement of the complex, object oriented approach.

Because object oriented modeling adheres closely to the concept of system development, it does not specifically focus on business issues. ARIS, however, was designed from the ground up to address business process issues, which is why it also includes business concepts such as production theory, activity based costing and enterprise organization in ARIS.

The views functions, organization, data, output and control compose an architectural concept which provides richer semantic functionality than the abstract system definition of object oriented models. This is due to the lack of a framework in these models, making it more difficult to recognize any overlapping or any contradictions within diagram types. Interestingly enough, up to eight different methods are used in the object oriented concept as well, despite its claims of a unified approach.

By no means, do we intend to pit ARIS against object oriented modeling. Quite the opposite, even when using object oriented modeling for specific system development, users should employ ARIS model views as well because of ARIS' greater focus on business issues and because ARIS model views are easier to understand.

F.II CIMOSA

The ESPRIT program, funded by the European Union (EU), has resulted in a series of research projects for developing an architecture, computer integrated manufacturing open system architecture (CIMOSA), for CIM systems. CIMOSA results have been published by several (groups of) authors, including *AMICE, CIMOSA 1993; Vernadat, Enterprise Modeling and Integration 1996 and* others. This project originally involved 30 participating organizations, including manufacturers as the actual users, IT vendors and research institutes. Although the project focused on CIM as an application goal, its mission was nevertheless to provide results for general enterprise modeling. One of CIMOSA's goals was also to provide an architecture and a methodology for vendor independent, standardized CIM modules to be "plugged" together, creating a customer oriented system („plug and play") *(see Vernadat, Enterprise Modeling and Integration 1996, p. 41).*

The CIMOSA modeling framework is based on the CIMOSA cube (see Fig. 81).

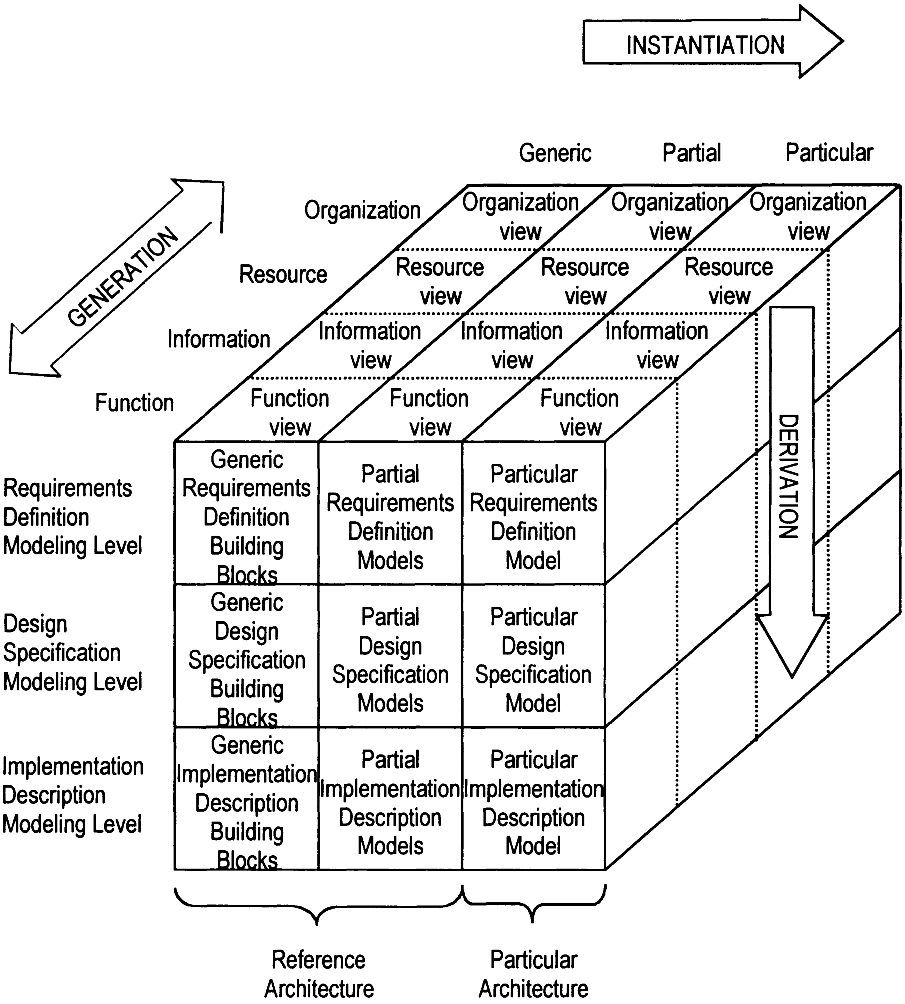


Fig. 81 The CIMOSA modeling architecture (CIMOSA cube)
(from Vernadat, Enterprise Modeling and Integration 1996, p. 45)

CIMOSA distinguishes three different dimensions, described by the three axes of the cube. The vertical direction ("stepwise derivation") describes the three description levels of the phase concept: requirements definition, design specification und implementation description. These levels are for the most part identical with those of the ARIS life cycle.

In the horizontal dimension ("stepwise instantiation"), concepts are individualized step by step. First, basic requirements (generic requirements, building blocks) are defined, then particularized in the next step according to industry specific

requirements (partial requirements). In step three, they are broken up into enterprise specific requirements (particular requirements).

This point of view makes clear that initially, according to CIMOSA, general building blocks should be used to define standards, after which the building blocks are grouped into industry specific reference models. In the last step, they are used for developing enterprise specific solutions. In ARIS, the degree of detailing an information model is defined while addressing the granularity issues.

By directly entering content related reference models, it becomes clear that the CIMOSA architecture combines general methodological issues regarding information systems and application related CIM domains.

The third dimension, "stepwise generation", describes the various views of an information system. This point of view has goals similar to ARIS regarding the creation of views, although not all the results are the same. CIMOSA divides description views into "function view", "information view", "resource view" and "organization view". "Function view" is the description of events, although it also includes a combination of other elements such as events and processes, including performance and exception handling *(see Vernadat, Enterprise Modeling and Integration 1996, p. 46)*. "Information view" refers to the data view or object definition. "Resource view" describes IT and production resources, "organization view" implies the hierarchical organization.

CIMOSA also breaks up the entire context into various views, although it lacks a level for reassembling them, as opposed to ARIS with its control and process views. This results in the fact that in CIMOSA, descriptions of the individual views are combined with one another. For example, when resources are being described, they are at the same time also allocated to functions. The CIMOSA modeling concept does not feature an output view.

The CIMOSA concept develops an architecture suitable for describing information systems, into which content in the form of standardized reference models, all the way to actual software generation, can be entered. Despite the above mentioned drawbacks it considers important points. Based on this concept, modeling methods are classified in CIMOSA and described by meta models, all the while adhering to an event driven, business process oriented view. Furthermore, enterprises are regarded as a series of multiple agents communicating with one another.

Despite the considerable financial and intellectual efforts spent on CIMOSA, its practical contribution so far has been minimal. Business users involved in the project have so far reported few special applications resulting therefrom, with the exception of the car manufacturer Renault with a repair service application for manufacturing plants, and the tooling company TRAUB AG with an application for optimizing individual development of tools. To date, a CIMOSA based modeling tool has not been used much in practice.

The main reason for the lack of success in real-world applications is presumably its very theoretical design, which does not incorporate commercially available IT solutions (standard software, for example). Considering the general lack of interest in CIM concepts, the extremely specialized focus of this approach seems to be working to its disadvantage.

F.III IFIP - Information System Methodology (ISM)

Olle et al. give a comprehensive methodology for developing more traditional information systems *(see Olle et al., Information Systems Methodologies 1991).* The designation "methodology" is used at the same level as the term "architecture". The seven authors of the study are members of the international federation for information processing (IFIP) task group, in particular, of the "design and evaluation of information systems" working group WG 8.1 of "information systems" technical committee TC 8. The research results of the study are summarized in the guideline "information systems methodology".

The design of the methodology does not focus on any particular IS development methods. Rather, it is based on a wide range of knowledge, including as many concepts as possible: IDA (interactive design approach), IEM (information engineering methodology), IML (inscribed high level Petri nets), JSD (Jackson system development), NIAM (Nijssen's Information Analysis Method), PSL/PSA (problem statement language/problem statement a analyser), SADT (structured analysis and design technique) as well as Yourdon's approach.

This methodology is described by meta models of an entity relationship concept. It features the point of view and stages of an information system life cycle, distinguishing data oriented, process oriented and behavior oriented perspectives (see Fig. 82). Creating these perspectives is less a matter of analytical conclusion than simply reflecting a goal of addressing the key issues typical in traditional IS developing methods *(see Olle et al., Information Systems Methodologies 1991, p. 12).*

Entity types and their attributes are reviewed in the data oriented perspective. The process oriented perspective describes events (business activities), including their predecessor or successor relationships. Events and their predecessor or successor relationships are analyzed in the behavior oriented perspective.

From a comprehensive 12-step life cycle model *(see Olle et al., Information Systems Methodologies 1991, p. 46)* we will select three steps: Information systems planning, business planning and system design and then examine the last two in detail, in terms of their key role in the methodology.

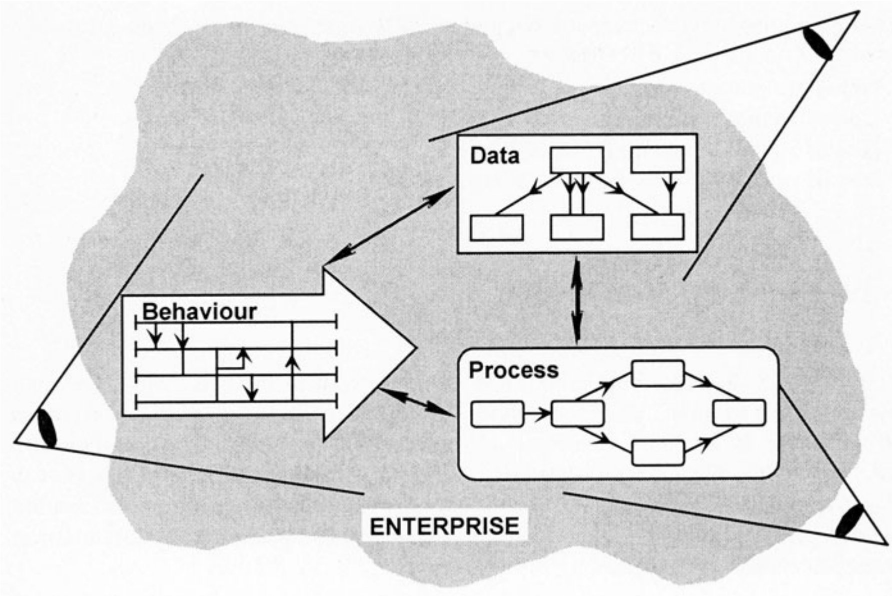


Fig. 82 Perspectives of the IFIP architecture
(from Olle et al., Information Systems Methodologies 1991, p. 13)

Information systems planning refers to the strategic planning of an information system. In business analysis, existing information systems of an entire enterprise or of a sub-area of the enterprise are analyzed. The respective information system is designed in the step system design. This concept also includes a comprehensive procedural model, including a role concept for project organization.

With regard to ARIS, this concept has some overlapping areas, in others there are deviations. What both concepts have in common is their two-dimensional point of view, with perspectives and development steps. There are differences in their instances, however. For example, Olle et al. do not explicitly list the organization view, but rather review it along with other activities, albeit rudimentarily. The process definition more or less dovetails with ARIS' function definition. Data and functions or events and functions are also strictly separated from one another. The three perspectives linked together are only slightly comparable with ARIS control view. The step system design blends together ARIS phases requirements definition and design specification, with the emphasis on the latter.

The main differences between ARIS and the IFIP model are the IFIP model's

* lack of an output view,
* lack of an organization view,
* lack of an implementation phase,
* lack of systematically addressed control view.

F.IV Zachman Framework

A framework for describing enterprises, quite popular in the U.S., was developed by J. A. Zachman *(see Zachman, Framework for Information Systems Architecture 1987; Sowa/Zachman, Extending and Formalizing the Framework for Information Systems Architecture 1992; Burgess/Hokel, Brief Introduction to the Zachman Framework 1994).* This concept is based on IBM's information systems architecture (ISA), but has been enhanced and presented by Zachman in numerous talks and seminars.

This approach (see Fig. 83) consists of 6 perspectives and 6 description boxes. In the ARIS terminology, Zachman's description boxes would equate to views and perspectives would equate to the levels of the life cycle model.

Perspectives are listed in brackets, along with the respective role designations of the party involved: scope (planner), enterprise model (owner), system model (designer), technology model (builder), components (sub-contractor) and functioning system (user).

The areas to be viewed are designated by interrogatives, with the respective actions listed in brackets: What (data), how (function), where (network), who (people), when (time) and why (rationale). Perspectives and files to be viewed are at a right angles to one other. Every field in the matrix is described by a method.

Contrary to ARIS, the Zachman framework is not capable of being directly implemented into an information system and the relationships between the description fields are not entered systematically. Furthermore, the relationship of Zachman's framework with the specific creation of output within the business process is not apparent.

Initial approaches for supporting tools are becoming apparent, thanks to cooperation with Framework Software, Inc., CA.

FOCUS

PERSPECTIVE

Generic Framework Element Bond Element	WHAT *(Dato)* Entity Relationship Entity	HOW *(Function)* Process Input-Output Process	WHERE *(Network)* Node Line Node	WHO *(People)* Agent Work Agent	WHEN *(Time)* Event Cycle Event	WHY *(Rationale)* End Means End
SCOPE *(Planner)*	Entity List	Process List	Location List	Organization List	Major Event List	Objective List
ENTERPRISE MODEL *(Owner)*	Enterprise Entity Enterprise Rule Enterprise Entity	Enterprise Process Resource Enterprise Process	Enterprise Location Enterprise Channel Enterprise Location	Organization Work Organization	Enterprise Event Enterprise Cycle Enterprise Event	Objective Strategy Objective
SYSTEM MODEL *(Designer)*	Entity Type Relationship Type Entity Type	System Process User View System Process	Site Link Site	Role Presentation Role	System Event System Cycle System Event	Criterion Choice Criterion
TECHNOLOGY MODEL *(Builder)*	Data Structure Referential Integrity Data Structure	Application Device Format Application	Connection Point Communication Line Connection Point	User Technical Interface User	Technical Event Technical Cycle Technical Event	Condition Action Condition
COMPONENTS *(Sub-contractor)*	Data Container Aquisition Data Container	Module/Object Couple/Message Module/Object	Address Protocol Address	Individual Transaction Individual	Component Event Component Cycle Component Event	Sub-condition Step/Task Sub-condition
FUNCTIONING SYSTEM *(User)*	Information Integrity Information	Procedure Request Procedure	Client/Server Access Client/Server	Worker Work Session Worker	Operating Event Operating Cycle Operating Event	Target Option Target

Fig. 83 Zachman architecture
(from Burgess/Hokel, Brief Introduction to the Zachman Framework 1994, p. 26)

F.V Research Results of the University of St. Gallen, Switzerland

In a series of research projects at the University of St. Gallen, Switzerland, various concepts for describing information systems were developed. This work ranges from procedural models and meta models to the definition of methods, from a method for business process design (PROMET) to a comparison of various methods and modeling tools. Although a specific recommendation for an architecture is not given, a framework from the requirements definition for evaluating the various BPR methods can be derived. Due to the fact that the method classification is based on this requirements definition, it is defined on the "logically higher level", so to speak, equating to the level of the ARIS concept *(see Bach/Brecht/ Hess/Österle, Enabling Systematic Business Change 1996, p. 38; Österle/Bremer/Hilbers, Unternehmensführung und Informationssystem 1992; Gutzwiller/Österle, Referenz-Meta-Modell Analyse 1990; Österle, Business Engineering I 1995, pp. 31).*

"Method components" and "design areas" are distinguished. Method components refer to the procedural model for BPR projects and are divided into the following components: Functions, organizational role concept, described results (deliverables), and techniques. "Design areas" correspond with the "views" that are to be described and are divided into: Workflow, process results (output), process management, information system, organizational structure and organizational culture.

This approach stresses in equal measure the procedural model and the results to be achieved. As opposed to ARIS, where the business process model is initially introduced as the key element for creating the views, individual design areas are not derived from a basic model. In ARIS, process output is described in the output view. For the most part, the ARIS procedural model addresses the "method components". In the ARIS HOBE concept, there is a tight integration between the concept and the workflow descriptions -- and the implementation of business processes into information systems, respectively. In ARIS, issues concerning organizational culture are addressed in the strategic planning stage.

In the St. Gallen studies, the lack of a specific architectural concept becomes apparent when the ERM objects of the "design area" which belong together are linked with the meta models by lines. In contrast, in the ARIS information model, the views for classifying the modeling objects have been defined from the very beginning.

F.VI Other Architectures

This list comprises only a fraction of the wide range of architectures. AD/CYCLE, announced by IBM and described in the first edition of this work, did not live up to expectations, although knowledge gained from it has been included in the follow-up product, AIX/CASE. Many concepts proposed by consultancies, software and hardware vendors have not lived up to expectations, e.g. James Martin's Information Engineering (IEM) concept *(see Martin, Information Engineering II 1990)*.

The Microsoft repository (MR), with an approach similar to AD/CYCLE, is being offered by Microsoft and several other software vendors. MR is a database for storing components, models and objects, along with their descriptions and relationships, ensuring reusability and open tool support. Evidently, MR is based on an open meta model (in accordance with the UML approach) and interfaces other repositories *(see Linthicum, Microsoft Repository 1997)*. Despite any reservations one might have toward this product, MR stands a better chance of success than AD/CYCLE.

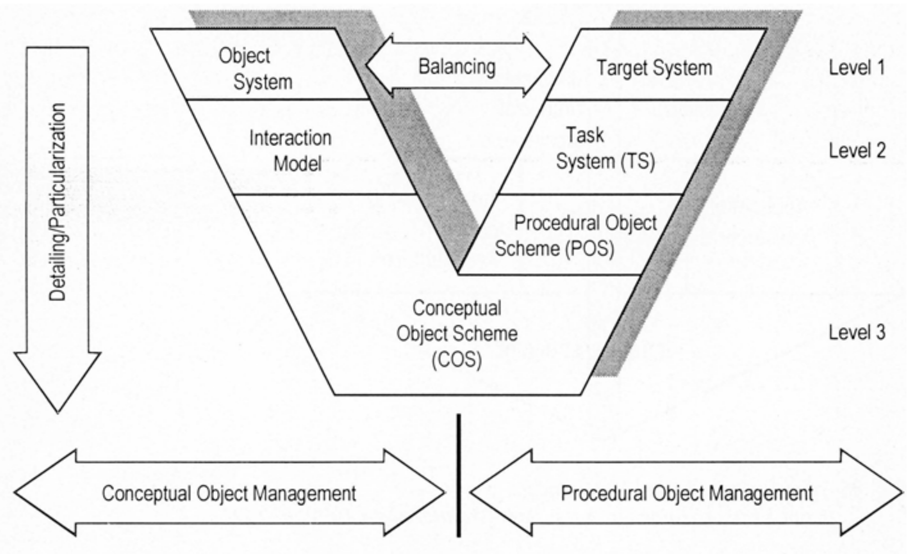


Fig. 84 Procedural model for object modeling in accordance with the SOM approach *(from Ferstl/Sinz, Wirtschaftsinformatik 1993, p. 137)*

With their semantic object model (SOM; *see Ferstl/Sinz, Vorgehensmodell zur Objektmodellierung 1993)*, Ferstl and Sinz presented a procedural model with a systematic structure of the description views by listing the results of the procedural model. The structure and behavior of the system are described in the SOM (see Fig. 84). Furthermore, a three-tier level concept is the basis for

detailing. Levels are continuously balanced between the structure model and the behavior model.

This model is enhanced by more detailed methods and example descriptions *(see Ferstl/Sinz, Wirtschaftsinformatik 1994)*. A prototype of a modeling tool is available. In contrast to ARIS, this concept is limited to a certain design method, the object oriented approach. It does not feature an independent organization and output view.

Krcmar's "information system architecture" (ISA, *see Krcmar, Informationssystem-Architekturen 1990*) is depicted as a "spinning top" (see Fig. 85). This is supposed to show that the structure is out of kilter if part of the description is missing. ISA stresses the link between the architecture and the enterprise strategy, as expressed by the vertical arrow. To date, this proposal has not be implemented in a real-world scenario.

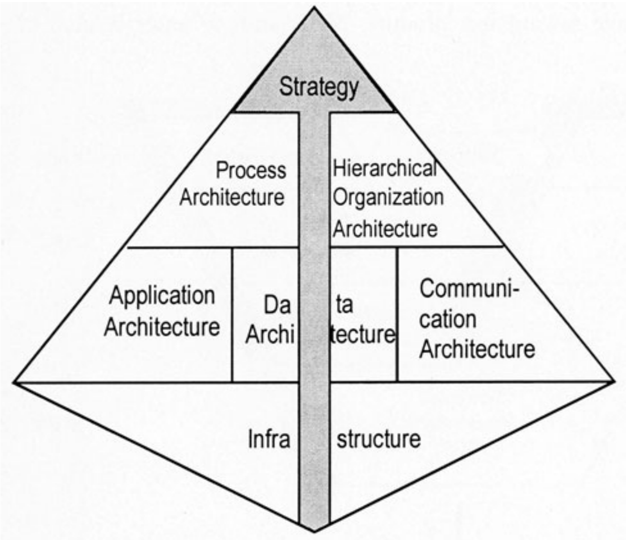


Fig. 85 Spinning top-shaped ISA concept
(from Krcmar, Informationssystem-Architekturen 1990, p. 399)

For other architecture approaches, see *Williams, Purdue Enterprise Reference Architecture 1991; Vernadat, Enterprise Modeling and Integration 1996; Sinz, Modellierung betrieblicher Informationssysteme 1996; Nüttgens, Koordiniert-dezentrales Informationsmanagement 1995; Oberweis, Modellierung von Workflows 1996; Donovan, Business Re-engineering 1994; Chen/Doumeingts, GRAI-CIM 1996; Bach/Brecht/Hess/Österle, Enabling Systematic Business Change 1996; Krcmar, Informationsmanagement 1997.*

G Deploying ARIS - Practical Procedures

We will now discuss a few selected applications of level I of the house of business engineering model, showing how to achieve a maximum real-world benefit from ARIS models. These applications are

* Business process reengineering,
* Quality certification according to ISO 9000 and
* Knowledge management.

The authors have practical expertise in these areas.

Additional success stories regarding the deployment of ARIS in

* Implementing standard applications (SAP R/3),
* Implementing workflow systems,
* Generating applications using frameworks and
* Modeling with UML

are given in *Scheer, ARIS - Business Process Modeling 1998.*

G.I ARIS Model Based Business Process Reengineering

Ralf Heib, Dipl.-Kfm., IDS Prof. Scheer GmbH, Saarbrücken, Germany

G.I.1 Process Oriented Enterprise Engineering

Previous business process reengineering approaches tended to focus on a radical new engineering of processes, regardless of the existing structures *(Hammer/Champy, Business Reengineering 1995)*. Other works stress the need for stepwise and continuous process improvement *(Harrington, Business Process Improvement 1991)*. What all these approaches have in common is the high priority of IT as an instrument for altering processes. In light of the wide range of possible optimization measures and their complexity, sound procedures and the deployment of appropriate methods and tools plays a major role.

In this book, we aim to describe a framework for business process optimization using the **ARIS procedural model**. This makes it possible to reengineer business

processes and to improve them continuously. Procedural models are based on a cyclical approach. New business processes are defined according to the analysis of existing structures (see Fig. 86), implemented using state-of-the-art IT, then regularly reviewed and modified as appropriate.

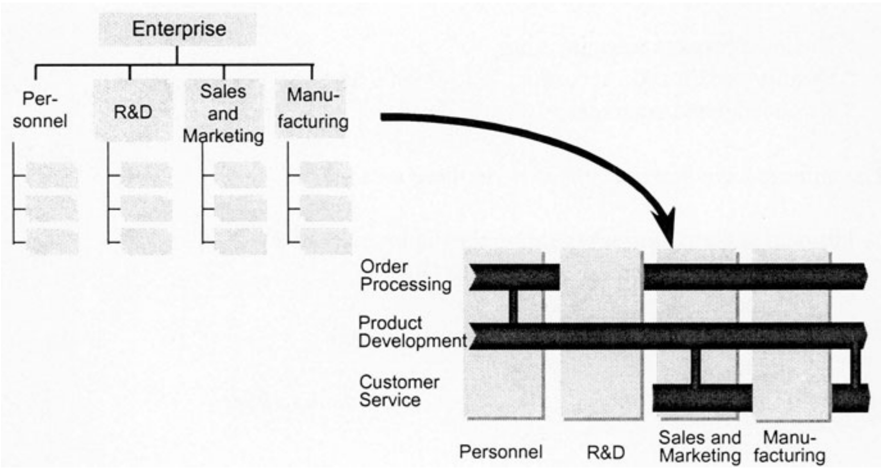


Fig. 86 Transition from function to process orientation

Enabling comprehensive and systematic description of business processes, their implementation as IT solutions and making it possible to view individual aspects of business processes such as organization, function and data structures, the ARIS method ensures integration of these individual aspects by using process views. Its standardized modeling methods increase transparency, enable project results to be compared and provide a common platform for discussing corporate processes for various hierarchies such as management, business users as well as organization and IT specialists.

One of the key advantages of the ARIS concept is its tight integration with the ARIS Toolset, increasing the efficiency of reengineering projects and ensuring reusability.

G.I.2 Procedural Model for Business Process Optimization

The procedural model presented here is based on the expertise of IDS Prof. Scheer GmbH, acquired in a wide range of business process optimization (BPO) projects. This model can be used as a checklist and guideline for BPO projects. The reference model is documented in the ARIS Toolset according to the ARIS methods, making it adaptable to each respective project situation. Project specific instances may be used like a "skills database" and then be applied to successive projects.

Key project phases are described by the value chain (see Fig. 87). Project activities and processes are additionally documented by EPCs, enhanced by organigrams (illustrations of an exemplary project organization) and data models (illustration of project results).

The individual project phases are as follows:

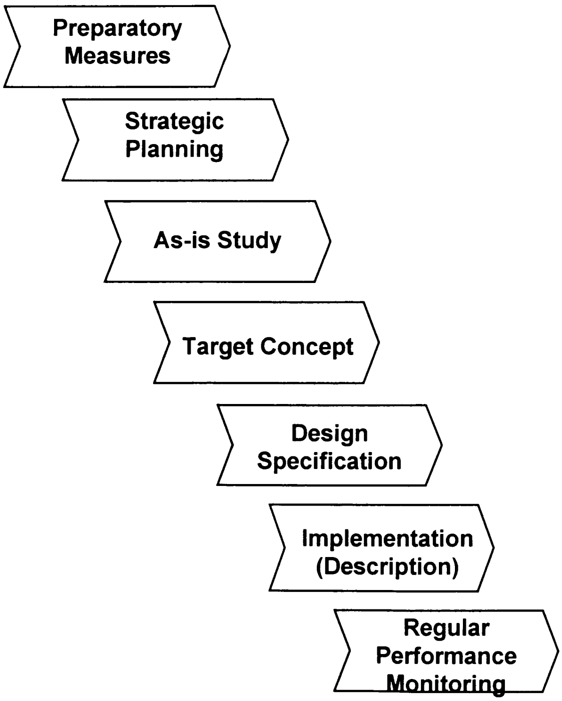


Fig. 87 Procedural model of business process optimization

G.I.3 Phases of Business Process Optimization

G.I.3.1 Preparatory Measures

Preparatory measures determine the outline of the project. During this phase, preliminary project goals are defined, project procedures are determined and project organization is modified accordingly. One of the key success factors in BPO projects is efficient execution of project organization. Management support should be ensured by a steering committee. Generally, project results are evaluated methodically and then consolidated by a core project team, in cooperation with IS or the organizational department, sometimes by third-party consultants. Process teams, made up of business users and members of the core project team, work on attaining business results.

Based on the project goals defined, the description methods are documented in a standards guide, on which project staff training is based, after which the project is presented to the employees in a kick-off event.

G.I.3.2 Strategic Planning

Optimizing business processes begins with a strategic positioning of the enterprise. Business processes should be engineered in such a way as to enable strategic implementation of corporate goals.

Strategic guidelines are established by product and output models and by target diagrams -- capturing key business areas of the enterprise, along with respective products, services and customer groups. Critical success factors and the target corporate hierarchy are modeled. Strategic guidelines are analyzed, particularizing BPO goals. Goals can be quantitative (reducing through-put time or cutting costs) -- or qualitative (increasing quality, flexibility or service quality). Project goals are documented in a target diagram.

G.I.3.3 As-is Study

As-is studies start with taking stock of the business processes. A framework architecture is created using value added process chains, documenting key business processes. This is the basis for more detailed description of the processes using the EPC method (see Fig. 88). In addition, the existing hierarchical organization is captured in organigrams, essential information objects in business term diagrams and existing application systems in application system diagrams.

Modeling business processes leads to transparency and identifies weak spots in processes as well as the potential for optimization. Current business processes are evaluated as to how well they meet BPO goals. Some criteria for evaluating business processes – and concepts for deriving potential for optimization – are: Through-put time (processing time, adjustment periods, delays or transport times), process costs, organizational hiatus (the amount of responsible entities in the pro-

cess), system hiatus (the amount of information systems in the process), media hiatus (the amount of shifts between manual and IS-supported processing), data redundancies and bottleneck situations in organizational units.

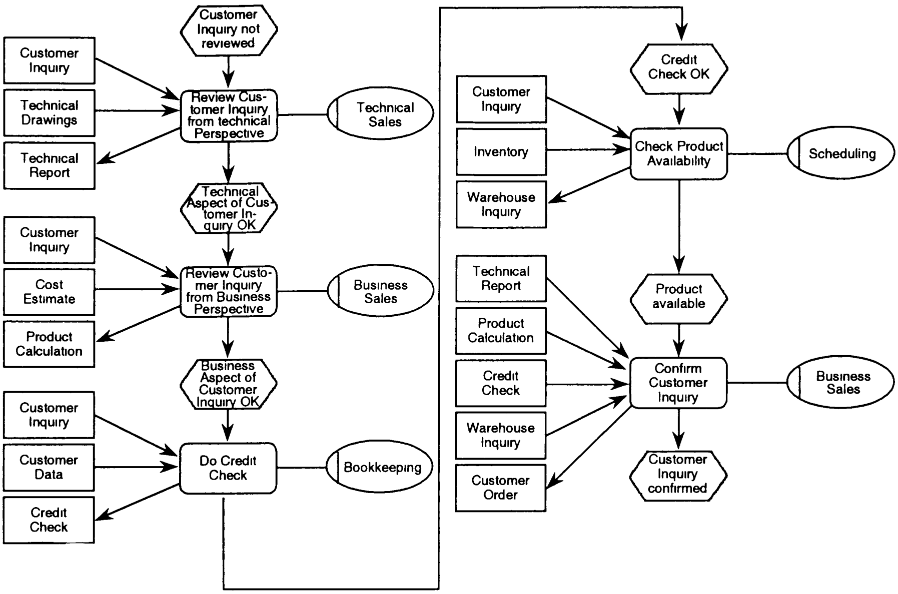


Fig. 88 EPC for business process "customer order processing"

Evaluating business processes and modeling the potential for optimization are key reengineering goals of the target concept.

G.I.3.4 Target Concept

Alternative target processes are defined in the target concept, starting with the examination of any weak spots in existing business processes. When engineering business processes, it is also possible to access reference models containing processes and organizational structures typical for the various vertical markets, based on expertise gained in comparable projects. Using reference models enables a quicker design of target processes because large portions of the reference structures can be adopted, freeing up project resources to focus on particular parts of the enterprise processes.

The resulting target processes are evaluated to determine how well goals have been met. ARIS Toolset supports the evaluation with its simulation and activity based costing tools, indicating how changes in the business process affect process

costs, through-put time, machine loads, etc. Hence, various "what if" scenarios might be simulated.
Based on the new target processes, the next step is to create an appropriate hierarchical organization in the form of an organigram. Organizational measures for ensuring new target processes should also be determined. This can include planning future personnel requirements or establishing the necessary qualification measures.

G.I.3.5 Design Specification

The design specification phase deals with planning how to implement target business processes by means of state-of-the-art IT systems, documented in an IT blueprint.

In each process area, the applications to be deployed are determined, such as custom-made solutions, standard applications, workflow, workgroup and document management programs or the Internet. The application actually selected depends on the business requirements defined in the target concept and enhanced by profitability evaluations -- and how well this particular application can be integrated with the corporate IT infrastructure. The resulting IT blueprint balances business processes, application systems and the IT infrastructure.

After the IT blueprint has been established, an implementation and migration plan which is the basis for implementing the business processes is worked-out. This plan determines an implementation strategy for the process areas and defines deadlines and resources for implementing the individual sub-projects.

G.I.3.6 Implementation

In the implementation phase, IT solutions for the various process areas are implemented. In parallel sub-projects, the target processes established are detailed further and then implemented into IT solutions. Software prototyping enables an early review of how well processes and IT solutions fit, ensuring user acceptance down the line.

G.I.3.7 Regular Monitoring and Continuous Process Improvement

After the implementation of business processes by deploying system solutions, the control and optimization phase is next. Now target processes and system solutions are reviewed as to how well they meet BPO goals. Key data for their evaluation can be derived directly from the IT systems (e.g., workflow systems delivering an analysis of through-put times, the cost of the supported processes and machine loads).

Regular performance monitoring leads to measures for customizing business processes and respective IT solutions, always with the goal of continuous process improvement in mind.

G.I.4 Summary

In light of the wide variety and complexity of possible BPO measures, sound procedures and the use of appropriate methods and tools play a key role in BPO projects. The ARIS procedural model for business process optimization is a concept that is not only methodically sound but also widely proven in practice. Close interaction of its procedures, method deployment and tool support ensure its ability to efficiently reengineer and customize business processes. The ARIS procedural model for business process optimization is the key to realizing quicker project through-put and an even higher standard of quality in project results. Its functionality for supporting methods and tools has the added benefit of cutting project costs as well. The resulting ARIS models represent an organizational knowledge base for customizing organization structures, enabling enterprises to meet new challenges.

G.II ARIS Model Based ISO 9000 Certification

Dr. Klaus Helling, IDS Prof. Scheer GmbH, Saarbrücken, Germany

G.II.1 ARIS Based Process Oriented Quality Management

Today, every company addressing the issue of quality management must familiarize itself with DIN EN ISO 900x standards (referred to in this text as ISO 900x). In this context, quality management systems ("QM system" in this text) include the organization structure, responsible entities, procedures, processes and the necessary means for realizing quality management. To date, real-world structuring of QM systems generally includes the 20 elements comprising ISO 9001, defining the basic requirements of a QM system in accordance with the standard. These 20 elements are shown in Fig. 89. Unfortunately, these systems are difficult for most users to understand because several requirements of various elements need to be considered for each activity, in which case it is almost impossible for employees to identify with the QM system.

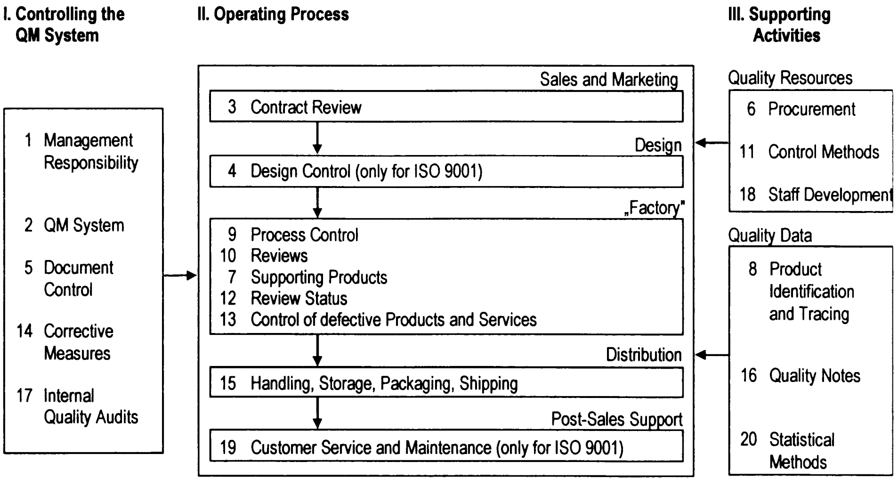


Fig. 89 Elements of ISO 9001
(from IM Magazin Qualität 1996, p. 66)

State-of-the-art strategies require quality management concepts that are consistently focused on business processes. This process oriented perspective makes it easier for employees to share in constructing the QM system because routine day-to-day tasks are described and the staff is not forced to use the abstract language of the standard. Involving the respective employees in the process of creating and updating detailed procedure and operating rules is essential. This improves user acceptance of the new QM system and also leverages employees' creativity in continuously improving and enhancing the QM system.

ARIS Toolset supports these strategies and helps to analyze, model and optimize all the processes relevant for ensuring a high standard of quality. This makes it an ideal tool for efficiently developing process oriented QM systems and doing ISO 900x certification prep work. ARIS Toolset also supports the structure of integrated management systems geared to the requirements of various standards such as ISO 9000, QS 9000, VDA Bd. 6, ISO 14000, EMAS, etc. *(see Helling/Herrmann, Änderungen flexibel meistern 1997)*. By focusing on relevant business processes, various content related requirements in quality, environment or security management can be displayed, documented and improved.

G.II.2 Procedural Model for ISO Certification

G.II.2.1 Procedural Model: An Overview

When designing quality management systems, it is crucial to involve every department and every employee in the process. Depending on the size of the company and the quality supporting management methods already in place, this takes 6 to 18 months. The key steps and milestones leading to certification are shown in the **procedural model for ISO certification** (see Fig. 90) featured in this work. We will discuss the steps involving the deployment of ARIS Toolset in greater detail.

G.II.2.2 Procedural Model: Benefits

Procedural models comprise comprehensive reference project processes, describing every step in the ARIS database, from strategic planning to attaining ISO certification. Procedural models combine the skills required to create process oriented QM documentation with hands-on instructions on how to deploy ARIS Toolset. By adding or deleting individual steps and by optionally allocating necessary resources to the individual steps, ARIS Toolset can be customized to meet corporate specifications. By linking the Toolset with project management systems, project plans can be generated directly from the procedural model. Procedural models reflect expertise gained in a large number of consulting projects and are a reliable guideline for ensuring certified QM systems and total quality management.

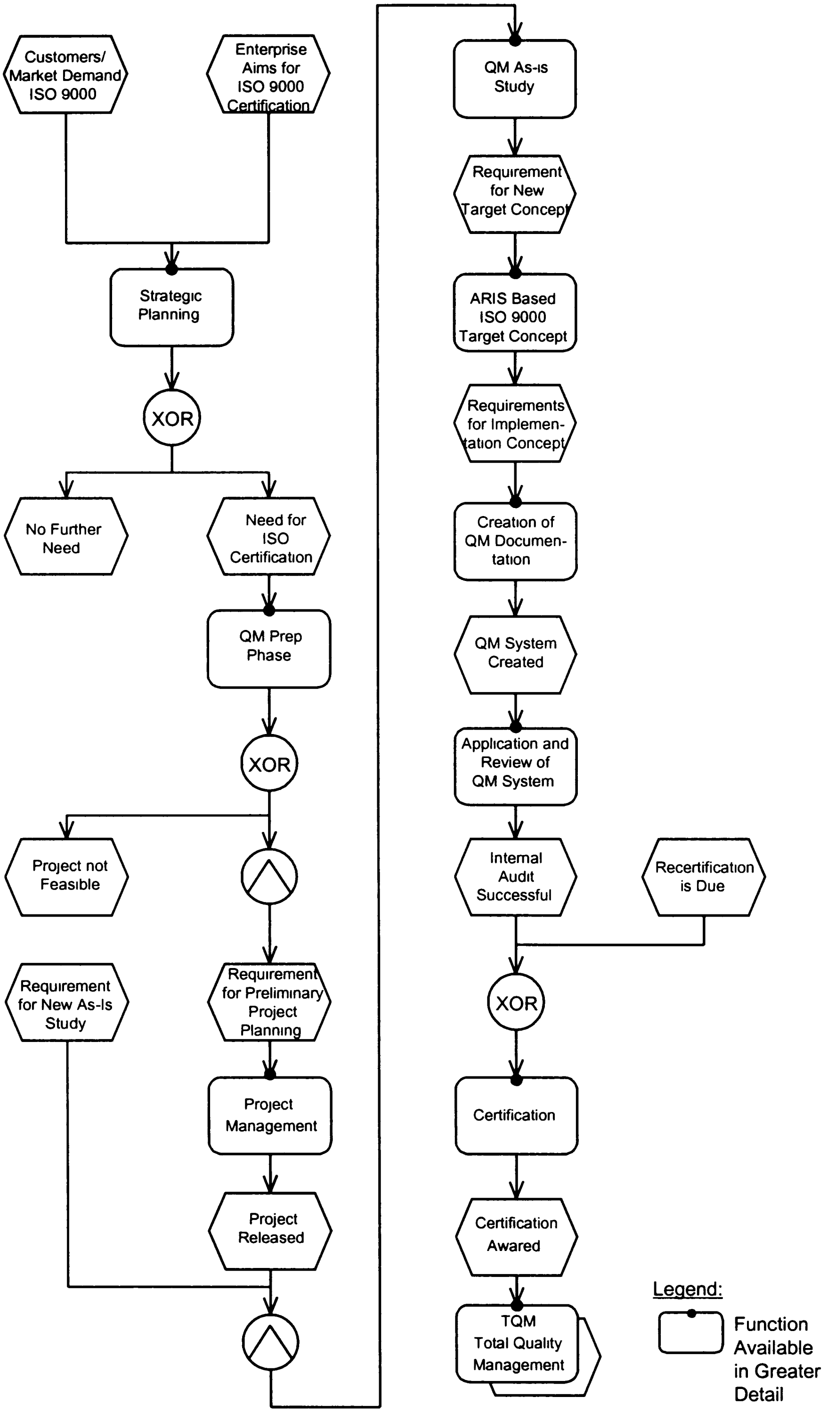


Fig. 90 Preliminary ARIS procedural model for ISO certification

G.II.3 Phases of the Procedural Model

The procedural model consists of eight essential phases, illustrated in Fig. 90 in an EPC and in Fig. 91 in a value chain. Every phase of the procedural model is described by a detailed EPC. If functions are available in greater detail by another EPC, the function's top border is marked by a dot. Furthermore, every process chain function is described verbally in more detail, including input and output data as well as the internal and third-party personnel involved. In level 3, every function requiring the deployment of ARIS Toolset is supplemented by another EPC, describing which ARIS Toolset methods and functions are required to support this task.

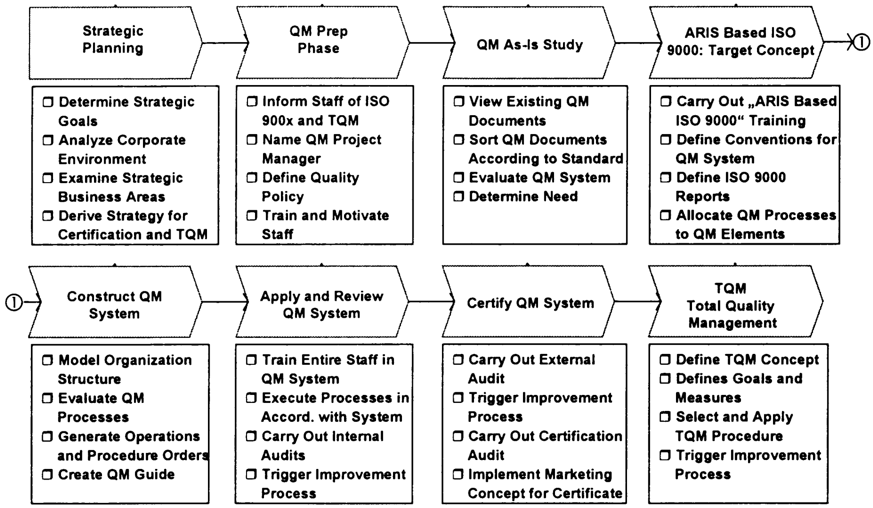


Fig. 91 Procedural model phases illustrated in a value chain

G.II.3.1 Strategic Planning

The purpose of strategic planning is to compile relevant strategic (long-term) corporate goals. These are defined after analyzing the strategic business areas and the corporate environment. Based on the strategic goals, measures for attaining these goals are outlined. It is critical for management to regard the construction of a quality management system as a strategic task. Thus, ARIS Toolset is deployed as a strategic instrument for documenting and continuously improving every enterprise process.

G.II.3.2 Prep Phase for Quality Management

QM prep phases usually consist of one or several workshops involving a company specific evaluation of the issues linked with ISO 900x certification. This also

includes a preliminary certification concept, naming a QM project manager and defining a corporate policy on quality. QM project managers are representatives of management, responsible for implementing the QS system. They must have access to top management and be able to act independently of line managers. If necessary, management can hire external consultants to help outline quality policies. However, specific definitions and commitments must come from the company itself, actually identifying with the quality goals. Appropriate strategies should be broadcast to the entire staff and the new quality goals should be supported wholeheartedly by everyone.

Finally, preliminary project procedures and the project management are outlined. Therefore, the ARIS procedural model might be modified to comply with ISO 9000 certification.

G.II.3.3 As-Is Study of the Quality Management System

Every enterprise has structures, rules and documents which lay the groundwork for smooth operation of corporate procedures and which should be integrated with the new QM system. As-is studies are the company specific analysis of the standard to be applied (ISO 9001, 9002 or 9003), reviewing which standard requirements regarding particular situations in the company are relevant to what extent. Stock is also taken of quality relevant documents and IS systems within the company.

The following questions should be answered in order to define which actions to take:

- Which existing processes can be used to create documentation complying with ISO 900x?
- Which existing processes need to be reworked?
- Which additional processes should be defined?

G.II.3.4 "ARIS Based ISO 9000": Target Concept

The methods used by ARIS Toolset are documented in a standards guide, containing an overview of key model types, object types, symbols and border definitions. The following ARIS model types at the requirements definition level are crucial when creating a QM system:

Organigrams, value chains and EPCs.

Business term models, information medium diagrams, node trees, information flow diagrams, process selection matrices, event chain diagrams and function allocation diagrams can also be used. Staff involved in the use of ARIS tools and methods obviously needs to be trained accordingly.

The preliminary process architecture is defined in the target concept phase, too. Identified processes can be allocated to the QM elements, providing an overview of how the various standard requirements are met by the processes.

G.II.3.5 Structuring the QM System

Process documentation is structured in accordance with the process architecture of the respective enterprise, processes being modeled top down by value chains. Across multiple levels, the value chains are broken up into their individual processes (see Fig. 92). The processes are then illustrated as EPCs. ARIS Toolset allows for a text description to be added to each EPC function. If quantitative process analyses are needed at a later point in time, processing times and costs can be entered. The responsible organizational entities, supporting documents, necessary proof documents and supporting applications are entered for every function, i.e., for every process step.

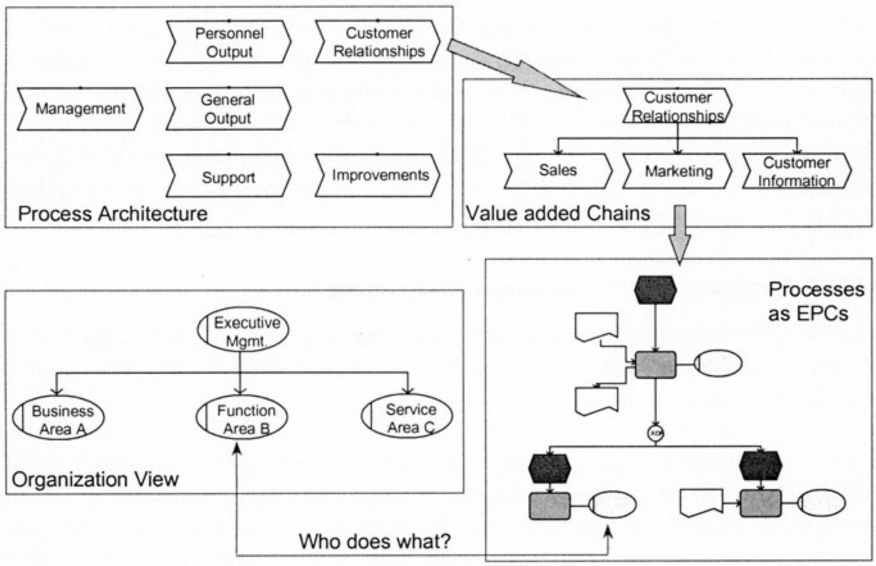


Fig. 92 ARIS models for quality management

In order to describe the organization view, the hierarchical organization is modeled using org charts. In addition to the hierarchical organization, role models are often used to structure QM systems. Various entities within the enterprise can assume roles, for example, quality managers, internal auditors, project managers, clerks or foremen. Organizational models are linked directly with the processes. This makes it immediately apparent who is responsible for which task. One should distinguish whether the respective organizational unit is responsible for a function, is executing the function, is involved in the function or only needs to be informed.

Job descriptions result directly from the allocation of positions or employees to the process chain functions.

Every person responsible for processes stores them in an enterprise-wide database using ARIS Toolset and is able to access current processes, valid organizational structures and documents. As soon as the documents are reconciled, their content is reviewed and then released. Then QM documentation is generated as a report, i.e., process overviews for the QM guide are created from the value chains in ARIS Toolset, and all the procedure instructions from the EPCs, respectively, are generated as well. Here, ARIS' advantages over traditional documentation -- where documents and cross references have to be maintained manually -- become apparent.

ARIS Toolset features process hierarchies in several steps, with cross references stored in the system ensuring a consistent process architecture.

Alternatively, processes can be directly put at the disposal of every person enterprise-wide involved in the project, leveraging the multi-user and network functionality of ARIS Toolset. There is no need to generate documentation and procedure instructions. The enterprise process model created with ARIS Toolset is also the valid documentation of the QM system. Assigning appropriate user and access privileges ensures that users have read access to the parts of the ARIS database relevant to them.

G.II.3.6 Applying and Reviewing QM Systems

Once the QM documentation is in effect, the processes are executed according to the new standards that have been documented. After an introductory phase, internal auditing of the QM system can begin.

A date for the internal audit must be agreed upon with the person in charge to be audited. That fact that an appointment is made demonstrates that internal audits are not some kind of surprise inspection for tracking down errors. Quite the contrary, their purpose is to coach these departments and to help them execute QM processes in accordance with the documentation. Audits also point out any errors in incorrect documentation.

The goal of internal audits is to improve processes. QM systems are not static but are subject to constant change. At various times and for various reasons it can be necessary to improve documentation, say, because content related or formalistic errors have been found or because QM processes have been altered, making otherwise current documentation obsolete. Improvements must obviously be documented as well. The fact that ARIS Toolset is based on a database capable of electronic delivery greatly simplifies updates and drastically reduces the subsequent effort of distributing the modified documentation.

Once the QM system has been successfully implemented and has been internally audited, this phase can be concluded. The enterprise is now ready for certification.

G.II.3.7 Certification

The exact procedure of this phase is based on contractual regulations with the respective certifier. Some key elements are filling out a short questionnaire, perhaps doing a pre-audit and of course actually auditing the enterprise. This process reviews whether regulations are meeting the standard requirements, whether the respective persons are aware of this and whether the regulations are actually being applied.

A QM certificate can be applied for once the internal audit has been carried out successfully. Certificates are valid for three years, with a certifier performing re-audits once a year, doing spot checks on the QM system. One focus of re-audits is to check whether the shortcomings of the previous audit have been corrected. Complete recertification every three years underscores the importance of always having QM documentation up-to-date.

G.II.3.8 Outlook and Framework: Total Quality Management

Being awarded a ISO 900x certificate does not mean that efforts regarding quality enhancement come to a standstill. Quality should constantly be measured in accordance with internal and external customers. Total quality management (TQM) requires process oriented ideas and operations. With process oriented QM system engineering a part of ISO 900x certifications, the "process concept" has become popular in enterprises. TQM philosophy requires constant improvement of corporate processes and the principle that existing conditions should always be questioned.

The duration and costs of every business process modeled with ARIS Toolset can be evaluated by ration analysis and simulation studies. ARIS Toolset's ABC (activity based costing) component controls processes value oriented. When processes are supported by a workflow system, the data resulting from the workflow system provides clues as to the efficiency of the processes. Actual costs and the duration of process execution indicate any potential for optimization.

G.III Using ARIS Models for Knowledge Management

Dr. Thomas Allweyer, IDS Prof. Scheer GmbH, Saarbrücken, Germany

G.III.1 Using Knowledge to Your Competitive Advantage

Despite the increasing importance of knowledge, according to estimates only about 30% of available organizational knowledge is actually utilized, with the remainder lying fallow *(see Zucker/Schmitz, Knowledge Flow Management 1994).* Studies show that most companies do not use their corporate knowledge correctly or to its full extent. For example, costly errors are frequently made because certain information is not available. Or critical knowledge is lost when certain individuals leave the company *(see Spek/Hoog, Knowledge Management 1994).* In recent years, middle management downsizing has frequently lead to a significant loss of knowledge. With corporate decentralization also leading to a decentralization of knowledge, making key information available across the enterprise has become a key issue. Otherwise, identical solutions for the same problems are invented over and over again.

In light of these challenges, more and more companies are starting to take appropriate measures to manage one of their key resources: information. Some companies have even created a new job position, chief knowledge officer *(see Davenport, Knowledge Management 1996)*, whose task it is to develop, monitor and improve strategies, processes, organization structures and technologies for processing corporate information *(see Probst/Raub/Rombardt, Wissen managen 1997).* Acquiring, displaying, delivering and utilizing knowledge depends on other corporate activities and even takes place within the business processes. Thus, understanding business processes is an important prerequisite for focused knowledge management.

In addition to activities that have always been knowledge-intensive such as product development or consulting, extremely standardized processes (order processing, for example) are becoming increasingly flexible in order to meet individual customers' demands better. This means that even more knowledge is now necessary to carry out these processes, requiring better trained and more experienced staff and the appropriate documents and handbooks, too.

Examining and improving business processes in order to process knowledge more effectively thus not only requires functions, data, organization and control flow to

be analyzed, but also makes it necessary to explicitly take knowledge into account – documented knowledge as well as the knowledge of the employees.

G.III.2 Knowledge Process Reengineering Procedures

Fig. 93 depicts a procedural model for improving the utilization of knowledge in the enterprise. Analogous to the term business process reengineering (BPR), we would like to suggest the term "knowledge process reengineering"

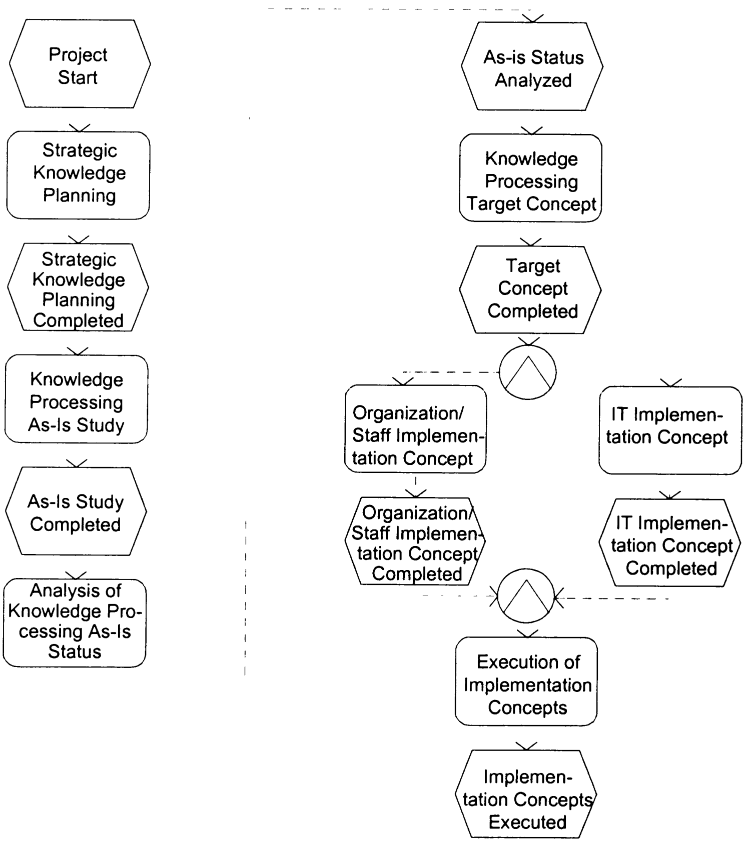


Fig. 93 Preliminary ARIS procedural model for knowledge process reengineering

The first step is to strategically plan knowledge. One of the key aspects of this step is to clarify what kind of knowledge is of key importance for the enterprise, so the project can be focused on these areas.

Step two is to analyze how knowledge is currently being processed, determining what knowledge in the enterprise is located where, how and where it is generated and how it is utilized.

The results of this as-is study are the foundation for analyzing how knowledge is processed, pointing out deficits and potential for improvement such as knowledge monopolies or poor utilization of existing knowledge.

When target concepts are created, this phase calls for specific process alterations to be defined, for example by modifying the documentation and distribution of knowledge within existing business processes, or by developing special "knowledge processes" for acquiring, preparing or distributing knowledge.

Next, these changes must be prepared by carrying out organizational and staff oriented measures like training and by preparing supporting information and communication technologies (groupware systems or intranets). Finally, these implementation concepts need to be implemented, staff needs to be trained and the new procedures need to be tested and further improved.

In the next chapter, we will discuss how these procedural model steps can be implemented by ARIS models.

G.III.3 The Phases of Knowledge Process Reengineering

G.III.3.1 Strategic Knowledge Planning

Based on strategic corporate goals, the main goal of strategic knowledge planning is to decide how these goals can be supported by knowledge management and which specific goals are a result of the project. For example, it could be strategically important for a company to become or to remain a leading technology provider. Thus, a knowledge management project should ensure that externally available knowledge regarding this technology is collected and taken into account. Furthermore, internal R&D in this field should be coordinated or perhaps stepped up. At any rate, it is especially important to improve the exchange, documentation of and easy access to knowledge in the enterprise.

This is why it is useful to illustrate the strategic target system and core corporate business processes as well as previously identified relevant knowledge categories in the ARIS Toolset and relate them to one another. These models are the starting point for detailed modeling in the following phase.

G.III.3.2 As-is Study of Knowledge Processing

In this project phase, the as-is status of the enterprise is captured and modeled. As previously mentioned, due to the close link between business processes and

knowledge management, it is first necessary to model business processes as EPCs. Frequently, such models already exist, having been created in business process reengineering or when implementing a standard software solution.

In order to illustrate knowledge processing, it is also important to model what knowledge categories are being processed, who possesses what kind of knowledge, what knowledge is necessary for which activities, where this knowledge is created or made available and so forth. In addition to the objects necessary for business process optimization, this also calls for additional object and model types. These are as follows (see the EPC illustration in ARIS Easy Design in Fig. 94):

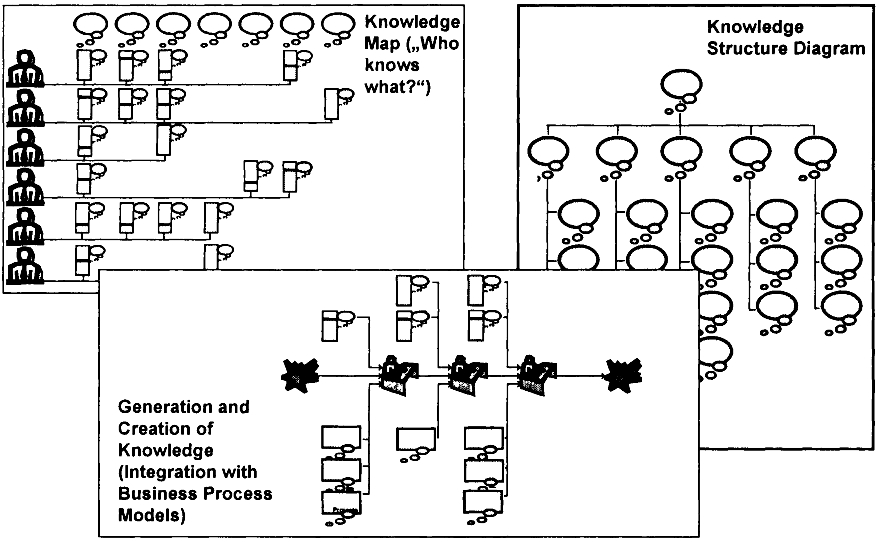


Fig. 94 Modeling the processing of knowledge

- **Knowledge structure diagrams**: Illustrate what kinds of knowledge are relevant for the enterprise. For example, knowledge necessary for executing a project could include, among other things, knowledge of the application area, knowledge of the procedure, project management methods and tools, demonstration and presentation skills, etc. We can distinguish between general, implicit knowledge of the employees and explicitly documented knowledge. The latter would include specific documents, files, application systems, etc. documenting the respective information.
- **Knowledge maps**: Document which employees or organizational units possess what kind of knowledge *(see Scheer/Bold/Hagemeyer/Kraemer, Informationssysteme im Wandel 1997)*. Various degrees of knowledge coverage can

also be depicted. These diagrams identify poorly covered knowledge areas or knowledge monopolies.
- **Generating and utilizing knowledge:** Here, additional information is modeled within existing business process models, illustrating for example to what extent what knowledge is necessary for carrying out a certain function, or what kinds of knowledge are explicitly documented when a function is executed.

Information and communication systems used in knowledge structure diagrams and business process models can be used for documenting, processing and distributing knowledge.

In addition to aspects that can only be depicted graphically, the description should also include models illustrating how knowledge is presented and prepared and whether there is a motivation for exchanging knowledge.

G.III.3.3 Analyzing the As-is Status

As-is modeling can provide specific benefits because it leads to documentation as to what knowledge is available at which location in the enterprise. This documentation makes it possible to quickly find a contact when particular questions arise.

A key benefit of models, however, is their capability to identify existing weak spots and potential for improvement in knowledge processing, such as:

- Strategically important areas of knowledge not covered by the enterprise,
- Knowledge monopolies potentially leading to the loss of critical knowledge, especially when certain individuals leave the company,
- Knowledge requirements that are not met,
- Knowledge available in the enterprise, but which is not utilized,
- Redundant acquisition and generation of the same knowledge,
- Useless or outdated knowledge profiles of employees,
- Rigid organizational separation of knowledge acquisition and utilization, impeding knowledge distribution,
- A lack of consistency in the IT and communication technology infrastructure actually designed to process knowledge.

G.III.3.4 Target Concept of Knowledge Processing

In accordance with the results of the analysis, in this stage, business processes are modified to improve knowledge processing. Among other things, this means that additional functions can be added to document the expertise and information gained and make it available to other employees.

Furthermore, specific knowledge processing procedures can be defined which would refresh, structure and distribute knowledge and finally remove outdated knowledge. For example, there could be a specific enterprise-wide process for collecting and compiling information on customer requirements and wishes that

sales staff could receive. This information could then be utilized for further product development and then distributed.

In addition, the necessary changes in organizational structures should be defined and knowledge profile targets for employees should be set. Last but not least, this step calls for the definition and modeling of requirements for appropriate support by IT and communication technology systems. It is important to ensure that knowledge management projects are driven by business requirements and not by technology. All too often, the only groupware component used in companies – despite implementations with the greatest of expectations – is e-mail. Or the only application running on the corporate intranet is the weekly company cafeteria menu. Problems such as these arise when specific user requirements are not focused on effective knowledge processing.

The target concept for knowledge processing is documented by the model types mentioned above.

G.III.3.5 Enterprise and Staff Implementation Concept

This is the phase where training concepts are developed and implemented to familiarize the staff with changes in the processes and with new information systems. As-is and target models developed in the previous phases are retrieved to determine which employees are affected by major changes and then used for specific measures and training classes. The models are also used for visualization and teaching purposes in class.

These knowledge profile targets are used for planning and executing qualification measures and when hiring personnel as well.

G.III.3.6 IT Implementation Concept

Target models also determine target requirements for IT and communication technology support (intranet solutions, groupware systems and document management systems, for example). It is usually necessary to integrate different systems and to make knowledge content regarding consistent distribution and delivery of knowledge accessible by means of a common structure such as Internet technology in intranets *(see Christmann-Jacoby/Maas, Wissensmanagement 1997)*.

IT implementation concepts also involve structuring contents, defining user interfaces, deploying specific services such as discussion groups or subscribing to information services. Tool based ARIS models, used in business processes and knowledge processing, are also used as process oriented structures for navigating knowledge contents. Starting with the models, the information required for the respective process is attained via hyperlinks.

G.III.3.7 Realizing Implementation Concepts

Realizing implementation concepts involves the respective training classes and qualification measures, preparing and controlling the modification of the business processes and organization structures as well as implementing IT and communication technology. Therefore, the ARIS models which have been developed provide the foundation.

The new processes and systems should be tested and corrected if applicable. Finally, enhancement of knowledge processing should be ensured by putting a continuous improvement process in place. A key prerequisite is to continuously update business process models and knowledge processing models in order to constantly ensure transparency of the current knowledge processing status *(see Allweyer, Adaptive Geschäftsprozesse 1997).*

References

Aichele, Kennzahlenbasierte Geschäftsprozeßanalyse 1997
Aichele, C.: Kennzahlenbasierte Geschäftsprozeßanalyse, Wiesbaden 1997.

Allweyer, Adaptive Geschäftsprozesse 1998
Allweyer, T.: Modellbasierte Gestaltung adaptiver Geschäftsprozesse, Wiesbaden 1998.

AMICE, CIMOSA 1993
ESPRIT Consortium Amice (Ed.): CIMOSA: Open System Architecture, 2. ed., Berlin et al. 1993.

Bach/Brecht/Hess/Österle, Enabling Systematic Business Change 1996
Bach, V., Brecht, L., Hess, T., Österle, H.: Enabling Systematic Business Change, Wiesbaden 1996.

Back-Hock, Executive-Information-System-Generatoren und –Anwendungen 1991
Back-Hock, A.: Unterstützung von Controlling-Aufgaben mit Executive-Information-System-Generatoren und –Anwendungen, in: Scheer, A.-W. (Ed.): Rechnungswesen und EDV, Tagungsband zur 12. Saarbrücker Arbeitstagung, Heidelberg 1991, pp. 36-59.

Balzert, Entwicklung von Software-Systemen 1992
Balzert, H.: Die Entwicklung von Software-Systemen - Prinzipien, Methoden, Sprachen, Werkzeuge, Mannheim et al. 1992.

Balzert, Lehrbuch der Software-Technik 1997
Balzert, H.: Lehrbuch der Software-Technik. Management, Qualitätssicherung, Querschnitte und Ausblicke. Unternehmensmodellierung, Heidelberg 1997.

Barker, Case Method 1990*
Barker, R.: Case* Method, Tasks and Deliverables, Wokingham et al. 1990.

Becker/Rosemann/Schütte, Grundsätze ordnungsgemäßer Modellierung 1995
Becker, J., Rosemann, M., Schütte, R.: Grundsätze ordnungsgemäßer Modellierung, in: Wirtschaftsinformatik 37(1995), pp. 435-445.

Binas-Holz, JAVA Programmierbuch 1996
Binas-Holz, A.: JAVA: Das Sprachkonzept, Programme und Applets; umfassende Beschreibung der APIs; Datenbankarbeit mit JAVA; die Werkzeuge des JAVA Developer Kits (JDK) und nützliche Add-Ons, Düsseldorf et al. 1996.

Boehm, Spiral Model 1988
Boehm, B.W.: A Spiral Model of Software Development and Enhancement, in: IEEE Computer 21(1988)5, pp. 14-24.

Brodie/Ridjanovic/Silva, Framework for Information Systems 1983
Brodie, M.L., Ridjanovic, D., Silva, E.O.: On a Framework for Information Systems Design Methodologies, in: Olle, T.W., Sol, H.G., Tully, C.J. (Ed.): Information Systems Design Methodologies: A Feature Analysis, Proceedings of the IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies, York, U.K., 5-7 July, 1983, Amsterdam et al. 1983.

Bröhl/Dröschel, V-Modell 1995
Bröhl, A.-P., Dröschel, W. (Ed.): Das V-Modell: Der Standard für die Softwareentwicklung mit Praxisleitfaden, 2. ed., München-Wien 1995.

Buck-Emden/Galimow, Client-Server-Technologie 1996
Buck-Emden, R., Galimow, J.: Die Client-Server-Technologie des SAP R/3-Systems, 3. ed., Bonn et al. 1996.

Bungert/Heß, Objektorientierte Geschäftsprozeßmodellierung 1995
Bungert, W., Heß, H.: Objektorientierte Geschäftsprozeßmodellierung, in: Information Management 10(1995)1, pp. 52-63.

Burgess/Hokel, Brief Introduction to the Zachman Framework 1994
Burgess, B.H., Hokel, T.A.: A Brief Introduction to the Zachman Framework, 2. ed., o. O. 1994.

Burt, OMG BOMSIG Survey 1995
Burt, C.: OMG BOMSIG Survey with published Definition of a Business Object, Revised version of the Business Object Management SIG survey, Document 95-02-04, 1995.

Casanave, Business-Object Architectures and Standards 1997
Casanave, C.: Business-Object Architectures and Standards, 1997, URL: http://www.tiac.net/users/oopsla/oo95wrkf.html.

Chen, Entity-Relationship Model 1976
Chen, P.P.: The Entity-Relationship Model: Towards a Unified View of Data, in: ACM (Ed.): Transactions on Database Systems, Nr. 1, 1976, pp. 9-36.

Chen/Doumeingts, GRAI-CIM 1996
Chen, D., Doumeingts, G.: The GRAI-CIM reference model, architecture and methodology, in: Bernus, P., Nemes, L., Williams, T.J. (Ed.): Architectures for Enterprise Integration, London et al. 1996, pp. 102-126.

Christmann-Jacoby/Maas, Wissensmanagement 1997
Christmann-Jacoby, H., Maas, R.: Wissensmanagement im Projektumfeld auf Basis von Internet-Technologien, in: IM Information Management 12(1997)3, pp. 16-26.

Coad/Yourdan, Object-Oriented Analysis 1991
Coad, P., Yourdan, E.: Object-Oriented Analysis, 2. ed., Englewood Cliffs 1991.

Coad/Yourdan, Object-Oriented Design 1991
Coad, P., Yourdan, E.: Object-Oriented Design, Englewood Cliffs 1991.

Codd, OLAP 1994
Codd, E.F.: OLAP – On-Line Analytical Processing mit TM/1, in: M.I.S. GmbH (Ed.), Darmstadt 1994.

Corsten, Dienstleistungsproduktion 1994
Corsten, H.: Dienstleistungsproduktion, SzU, Band 52, Wiesbaden 1994.

Davenport, Knowledge Management 1996
Davenport, T.: Some Principles of Knowledge Management, Austin, Texas 1996, URL: http://www.bus.utexas.edu/kman/kmprin.htm.

Donovan, Business Re-engineering 1994
Donovan, J.J.: Business Re-engineering with Information Technology, Englewood Cliffs 1994.

Engels, OLAP 1995
Engels, E.J.: On-line Analytical Processing – Neue Analysesystematik für die Datenbasis, in: Controlling 7(1995)2, pp. 98-105.

Eversheim, Prozeßorientierte Unternehmensorganisation 1994
Eversheim, W. (Ed.): Prozeßorientierte Unternehmensorganisation - Konzepte und Methoden zur Gestaltung "schlanker" Organisationen, Berlin et al. 1994.

Farny, Produktions- und Kostentheorie der Versicherung 1965
Farny, Produktions- und Kostentheorie der Versicherung, Karlsruhe 1965.

Ferstl/Sinz, SOM 1993
Ferstl, O., Sinz, E.J.: Vorgehensmodell zur Objektmodellierung, München 1993.

Ferstl/Sinz, Wirtschaftsinformatik 1993 bzw. 1994
Ferstl, O., Sinz, E.J.: Grundlagen der Wirtschaftsinformatik, 1. ed., München 1993 und 2. ed., München 1994.

Fingar/Read/Stickeleather, Next Generation Computing 1996
Fingar, P., Read, D., Stickeleather, J.: Next Generation Computing. Distributed Objects for Business, New York 1996.

Fingar, Blueprint for Business Objects 1996
Fingar, P.: The Blueprint for Business Objects, New York 1996.

Finkeißen/Forschner/Häge, Werkzeuge zur Prozeßanalyse 1996
Finkeißen, A., Forschner, M., Häge, M.: Werkzeuge zur Prozeßanalyse und -optimierung, in: Controlling, Heft 1/1996, pp. 58-67.

Floyd, Systematic Look at Prototyping 1984
Floyd, C.: A Systematic Look at Prototyping, in: Budde et al. (Ed.): Approaches to Prototyping, Berlin et al. 1984.

Fosdick, Ten Steps to AD/CYCLE 1990
Fosdick, H.: Ten Steps to AD/CYCLE, in: Datamation, Dezember 1, 1990, pp. 59-64.

Frank, Multiperspektivische Unternehmensmodellierung 1994
Frank, U.: Multiperspektivische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung, München et al. 1994.

Frese, Grundlagen der Organisation 1995
Frese, E.: Grundlagen der Organisation, Konzept - Prinzipien - Strukturen, 6. ed., Wiesbaden 1995.

Gaitanides, Prozeßorganisation 1983
Gaitanides, M.: Prozeßorganisation: Entwicklung, Ansätze u. Programme prozeßorientierter Organisationsgestaltung, München 1983.

Galler, Vom Geschäftsprozeßmodell zum Workflow-Modell 1997
Galler, J.: Vom Geschäftsprozeßmodell zum Workflow-Modell, Wiesbaden 1997.

Glaser, Prozeßkostenrechnung 1992
Glaser, H.: Prozeßkostenrechnung - Darstellung und Kritik, in: ZfbF, 44 (1992), pp. 275-288.

Glaser/Geiger/Rohde, PPS 1992
Glaser, H., Geiger, W., Rohde, V.: PPS - Produktionsplanung und -steuerung, 2. ed., Wiesbaden 1992.

Goldammer, HTML-Script ruft Java-Applet 1996
Goldammer, G.: HTML-Script ruft Java-Applet. Eine neue Entwicklungstechnik unter software- und anwendungstechnologischem Aspekt, in: IM Information Management 11 (1996) 3, pp. 6-14.

Gutenberg, Die Produktion 1983
Gutenberg, E.: Die Produktion, Grundlagen der Betriebswirtschaftslehre, Bd. 1: Die Produktion, 24. ed., Berlin et al. 1983.

Gutzwiller/Österle, Referenz-Meta-Modell Analyse 1990
Gutzwiller, T., Österle, H.: CC RIM, Referenz-Meta-Modell Analyse, Forschungsbericht Nr. IM 2000/CCRIM/2, Version 2.0, Institut für Wirtschaftsinformatik, Hochschule St. Gallen, 15.01.1990.

Hagedorn/Bissantz/Mertens, Data Mining 1997
Hagedorn, J., Bissantz, N., Mertens, P.: Data Mining (Datenmustererkennung): Stand der Forschung und Entwicklung, in: Wirtschaftsinformatik 39(1997)6, pp. 601-612.

Hagemeyer/Rolles/Schmidt/Scheer, Arbeitsverteilungsverfahren 1998

Hagemeyer, J., Rolles, R., Schmidt, Y., Scheer, A.-W.: Arbeitsverteilungsverfahren in Workflow-Management-Systemen: Anforderungen, Stand und Perspektiven, in: Veröffentlichungen des Instituts für Wirtschaftsinformatik, Saarbrücken 1998, Nr. 144.

Hammer/Champy, Business Reengineering 1995

Hammer, M., Champy, J.: Business Reengineering: Die Radikalkur für das Unternehmen, 5. ed., Frankfurt/Main-New York 1995.

Hars, Referenzdatenmodelle 1994

Hars, A.: Referenzdatenmodelle - Grundlagen effizienter Datenmodellierung, Wiesbaden 1994.

Harrington, Business Process Improvement 1991

Harrington, H. J.: Business Process Improvement, New York et al. 1991.

Hildebrand, Software Tools 1990

Hildebrand, K.: Software Tools: Automatisierung im Software-Engineering - Eine umfassende Darstellung der Einsatzmöglichkeiten von Software-Entwicklungswerkzeugen, Berlin et al. 1990.

Hoff/Shaio/Starbuck, Java-Applets 1996

Hoff, A. v., Shaio, S., Starbuck, O.: Java-Applets erstellen und nutzen, Bonn 1996.

Hofinger, IBM Repository Manager 1991

Hofinger, A.: Der IBM Repository Manager 1991, in: HMD 161/1991, pp. 45-54.

Hollingsworth, Workflow Reference Model 1995

Hollingsworth, D.: The Workflow Reference Model, in: Workflow Management Coalition (Ed.): Document TC00-1003, Draft 1.1, 1995.

Horton, Information Management Workbook 1981

Horton, F.W.: The Information Management Workbook - IRM made simple, Washington DC 1981.

Horváth/Mayer, Prozeßkostenrechnung 1989

Horváth, P., Mayer, R.: Prozeßkostenrechnung, in: Controlling, 1 (1989) 4, pp. 214-219.

IDS, ARIS-Toolset 1997

IDS Prof. Scheer GmbH (Ed.): Produktinformation ARIS-Toolset, Saarbrücken 1997.

IDS, ARIS-Easy Design 1997

IDS Prof. Scheer GmbH (Ed.): Produktinformation ARIS-Easy Design, Saarbrücken 1997.

IDS, Workflow-Modellierung 1997

IDS Prof. Scheer GmbH (Ed.): White Paper Workflow-Modellierung, Vers. 1.1, Saarbrücken 1997.

IBM, Shareable Frameworks 1996

IBM AG (Ed.): Shareable Frameworks Technical Overview, 1996, URL: http://www.softmall.com/sf/technov.html.

Imai, Kaizen 1992

Imai, M.: Kaizen, Der Schlüssel zum Erfolg, Der Japaner im Wettbewerb, München 1992.

IM Magazin Qualität 1996

IM Magazin Qualität, in: IM Information Management 11(1996)4, pp. 66-69.

Inmon, Data Warehouse 1993

Inmon, W.H.: Building the Data Warehouse, New York–Chichester-Brisbane 1993.

Jacobsen, Object-Oriented Software Engineering 1996

Jacobsen, I.: Object-Oriented Software Engineering. A Use Case Driven Approach, Wokingham 1996.

Jahnke/Groffmann/Kruppa, OLAP 1996

Jahnke, B., Groffmann, H.-D., Kruppa, S.: On-Line Analytical Processing (OLAP), in: Wirtschaftsinformatik 38(1996)3, pp. 321-324.

Jost, EDV-gestützte CIM-Rahmenplanung 1993

Jost, W.: EDV-gestützte CIM-Rahmenplanung, Wiesbaden 1993.

Kagermann, Business Framework 1997

Kagermann, H.: Buildung Industry Solutions within the Business Framework, Vortrag Sapphire, Amsterdam Juni 1997, URL: http://www.sap.com/events/amsterdam/index.htm.

KBSt, V-Modell 1992

KBSt (Ed.): Planung und Durchführung von IT-Vorhaben in der Bundesverwaltung - Vorgehensmodell (V-Modell), Schriftenreihe der KBSt, Band 27/1, August 1992.

Keller/Nüttgens/Scheer, Semantische Prozeßmodellierung 1992

Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung, in: Veröffentlichungen des Instituts für Wirtschaftsinformatik, Saarbrücken 1992, Nr. 89.

Keller/Schröder, Geschäftsprozeßmodelle 1996

Keller, G., Schröder, G.: Geschäftsprozeßmodelle: Vergangenheit - Gegenwart - Zukunft, in: Management & Computer 4(1996)2, pp. 77-90.

Keller/Teufel, SAP R/3 prozeßorientiert anwenden 1997

Keller, G., Teufel, T.: SAP R/3 prozeßorientiert anwenden: Iteratives Prozeß-Prototyping, Bonn et al. 1997.

Kern, Industrielle Produktionswirtschaft 1992

Kern, W.: Industrielle Produktionswirtschaft, 5. ed., Stuttgart 1992.

Kirchmer, Implementation of Standard Software 1998
Kirchmer, M.: Business Process Oriented Implementation of Standard Software, Berlin-Heidelberg 1998.

Kloock, Prozeßkostenrechnung 1992
Kloock, J.: Prozeßkostenrechnung als Rückschritt und Fortschritt der Kostenrechnung, in: Kostenrechnungspraxis, 5 (1992), pp. 183-193.

König/Packowski/Wyler, BPR als Chance 1995
König, A., Packowski, J., Wyler, S.: BPR als Chance für prozeßorientierte QM-Systeme, in: QZ 40 (1995) 10, pp. 1204-1208.

Kraemer, Kostenmanagement 1993
Kraemer, W.: Effizientes Kostenmanagement – EDV-gestützte Datenanalyse und –interpretation durch den Controlling-Leitstand, Wiesbaden 1993.

Krcmar, Informationssystem-Architekturen 1990
Krcmar, H.: Bedeutung und Ziele von Informationssystem-Architekturen, in: Wirtschaftsinformatik, 32 (1990) 5, pp. 395-402.

Krcmar, Informationsmanagement 1997
Krcmar, H.: Informationsmanagement, Berlin et al. 1997.

Krumbiegel, Integrale Gestaltung von Geschäftsprozessen 1997
Krumbiegel, J.: Integrale Gestaltung von Geschäftsprozessen und Anwendungssystemen in Dienstleistungsbetrieben, Wiesbaden 1997.

Kruppke, Software 2000 1997
Kruppke, H.: Software 2000 - Modellierung und Generierung von individuellen Business Lösungen aus Basiskomponenten, Saarbrücken 1997.

Kueng/Bichler/Schrefl, Geschäftsprozeßmodellierung 1996
Kueng, P., Bichler, P., Schrefl, M.: Geschäftsprozeßmodellierung: Ein zielbasierter Ansatz, in: IM Information Management 11(1996)2, pp. 40-50.

Küting, Benchmarking von Geschäftsprozessen 1996
Küting, K.: Benchmarking von Geschäftsprozessen als Instrument der Geschäftsprozeßanalyse, in: Berkau, C., Hirschmann, P. (Ed.): Kostenorientiertes Geschäftsprozeßmanagement, München 1996, pp. 121-140.

Lang, Gestaltung von Geschäftsprozessen 1997
Lang, K.: Gestaltung von Geschäftsprozessen mit Referenzbausteinen, Wiesbaden 1997.

Linthicum, Microsoft Repository 1997
Linthicum, D.S.: Microsoft Repository I.O., in: DBMS, Juni 1997, pp. 28-32.

Lockemann/Dittrich, Architektur von Datenbanksystemen 1987
Lockemann, P.C., Dittrich, K.R.: Architektur von Datenbanksystemen, in: Lockemann, P.C., Schmidt, J.W. (Ed.): Datenbank-Handbuch, Berlin et al. 1987.

Long, Taxonomy of BPR Tools 1992
Long, K.: A Taxonomy of BPR Tools and Techniques, in: Proceedings of the National Business Process Reengineering Conference, Washington DC 1992.

Maier, Qualität von Datenmodellen 1996
Maier, R.: Qualität von Datenmodellen, Wiesbaden 1996.

Martin, Application Development 1982
Martin, J.: Application Development Without Programmers, Englewood Cliffs 1982.

Martin, Information Engineering I 1989
Martin, J.: Information Engineering, Book I: Introduction, Englewood Cliffs 1989.

Martin, Information Engineering II 1990
Martin, J.: Information Engineering, Book II: Planning and Analysis, Englewood Cliffs 1990.

Meinhardt, Geschäftsprozeßorientierte Einführung von Standard-Software 1995
Meinhardt, S.: Geschäftsprozeßorientierte Einführung von Standard-Software am Beispiel des SAP-Systems "R/3", in: Wirtschaftsinformatik 37(1995)5, pp. 487-499.

Meyer, Object-oriented Software Construction 1988
Meyer, B.: Object-oriented Software Construction, New York 1988.

Muksch/Behme, Data Warehouse-Konzept 1997
Muksch, H., Behme, W.: Das Data Warehouse-Konzept, Architektur - Datenmodelle - Anwendungen, 2. ed., Wiesbaden 1997.

Muksch/Holthuis/Reiser, Data Warehouse-Konzept 1996
Mucksch, H., Holthuis, J., Reiser, M.: Das Data Warehouse Konzept – ein Überblick, in: Wirtschaftsinformatik 38 (1996) 4, pp. 421-433.

Müller, Informationsprodukte 1995
Müller, W.: Informationsprodukte, Ein Beschreibungsmodell für Versicherungsprodukte und seine Überprüfung anhand von Allgemeinen Versicherungsbedingungen, in: ZfB 65 (1995) 9, pp. 1017-1044.

Myers, Knowledge Management and Organizational Design 1996
Myers, P.S.: Knowledge Management and Organizational Design, Boston et al. 1996.

Nippa/Picot, Prozeßmanagement und Reengineering 1996
Nippa, M., Picot, A. (Ed.): Prozeßmanagement und Reengineering: Die Praxis im deutschsprachigen Raum, 2. ed., Frankfurt 1996.

Nüttgens, Koordiniert-dezentrales Informationsmanagement 1995
Nüttgens, M.: Koordiniert-dezentrales Informationsmanagement, Wiesbaden 1995.

Oberweis, Modellierung von Workflows 1996
Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen, Stuttgart et al. 1996.

Oestereich, Objektorientierte Softwareentwicklung 1997
Oestereich, B.: Objektorientierte Softwareentwicklung, München et al. 1997.

Olle et al., Information System Methodologies 1991
Olle, T.W. et al.: Information System Methodologies: A Framework for Understanding, 2. ed., Wokingham et al. 1991.

Olle/Sol/Tully, Information Systems Design Methodologies 1983
Olle, T.W., Sol, H.G., Tully, C.J. (Ed.): Information Systems Design Methodologies: A Feature Analysis, Proceedings of the IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies, Amsterdam et al. 1983.

Olle/Verrijn-Stuart/Bhabuta, Information Systems Life Cycle 1988
Olle, T.W., Verrijn-Stuart, A.A., Bhabuta, L. (Ed.): Computerized Assistance During The Information Systems Life Cycle, Proceedings of the IFIP WG 8.1 Working Conference on Computerized Assistance during the Information Systems Life Cycle, CRIS 88, Amsterdam et al. 1988.

OMG, Common Business Objects 1996
Object Management Group (Ed.): Common Business Objects and Business Object Facility, OMG TC Document CF/96-01-04,
URL: http://dataaccess.com/bodtf/Download/CFRFP4.doc.

Österle/Bremer/Hilbers, Unternehmensführung und Informationssystem 1992
Österle, H., Bremer, W., Hilbers, K.: Unternehmensführung und Informationssystem - Ansatz des St. Galler Informationssystem-Managements, 2. ed., Stuttgart 1992.

Österle, Business Engineering 1 1995
Österle, H.: Business Engineering. Prozeß- und Systementwicklung. Band 1: Entwurfstechniken, Berlin et al. 1995.

Österle, Business Engineering 2 1995
Österle, H.: Business Engineering. Prozeß- und Systementwicklung. Band 2: Fallbeispiel, Berlin et al. 1996.

Pentland, Process Grammars 1994
Pentland, B.T.: Process Grammars; A Generative Approach to Process Redesign, Document Draft, Los Angeles 1994, URL: http://ccs.mit.edu/CCSWP178/CCSWP178.html.

Petersohn, Klassifikation bei Entscheidungsproblemen 1997
Petersohn, H.: Vergleich von multivariaten statistischen Analyseverfahren und Künstlichen Neuronalen Netzen zur Klassifikation bei Entscheidungsproblemen in der Wirtschaft, Frankfurt am Main 1997.

Pietsch, Kundenorientiertes Softwareprozeßmanagement 1997
Pietsch, W.: Kundenorientiertes Softwareprozeßmanagement, in: IM Information Management 12(1997)1, pp. 21-26.

Plattner, Products & Organization 1997
Plattner, H.: Components: Products & Organizations. An Integrated and Innovative Approach to Technology and Organization for Business Solutions for Your Next Millenium, Vortrag Sapphire, Orlando August 1997, URL: http://www.sap.com/events/amsterdam/index.htm.

Pree, Komponentenbasierte Softwareentwicklung 1997
Pree, W.: Komponentenbasierte Softwareentwicklung mit Frameworks, Heidelberg 1997.

Preßmar/Eggers/Reinken, Interaktive Entwurfsmethode 1989
Preßmar, D.B., Eggers, S., Reinken, W.: Interaktive Entwurfsmethode zur computergestützten Herstellung betriebswirtschaftlicher Anwendungssoftware, in: Kurbel, K., Mertens, P., Scheer, A.-W. (Ed.): Interaktive betriebswirtschaftliche Informations- und Kommunikationssysteme, Berlin-New York 1989.

Probst/Raub/Rombardt, Wissen managen 1997
Probst, G., Raub, S., Rombardt, K.: Wissen managen: Wie Unternehmer ihre wertvollste Ressource optimal nutzen, Wiesbaden 1997.

Reichmann, Controlling mit Kennzahlen 1997
Reichmann, T.: Controlling mit Kennzahlen und Managementberichten, 5. Auflage, München 1997.

Reiter/Wilhelm/Geib, Multiperspektivischen Informationsmodellierung 1997
Reiter, C., Wilhelm, G., Geib, T.: Toolunterstützung bei der multiperspektivischen Informationsmodellierung, in: Management & Computer 5 (1997)1, pp. 5-10.

Remme, Konstruktion von Geschäftsprozessen 1997
Remme, M.: Konstruktion von Geschäftsprozessen, Wiesbaden 1997.

Remme, Organisationsplanung 1997
Remme, M.: Organisationsplanung durch konstruktivistische Modellierung, in: Veröffentlichungen des Instituts für Wirtschaftsinformatik, Saarbrücken 1997, Nr. 137.

Rosemann, Komplexitätsmanagement in Prozeßmodellen 1996
Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen - Methodenspezifische Gestaltungsempfehlungen für die Informationstechnologie, Wiesbaden 1996.

Rumbaugh et al., Object-Oriented Modeling and Design 1991
Rumbaugh, J. et al.: Object-Oriented Modeling and Design, Englewood Cliffs 1991.

SAP, White Paper Business Framework 1996
SAG AG (Ed.): White Paper Business Framework, Walldorf 1996, URL: http://www.sap-ag.de/bfw/media/pdf/50016528.pdf.

SAP, White Paper Business Objects 1997
SAP AG (Ed.): White Paper Business Objects, Walldorf 1997, URL: http://www.sap-ag.de/bfw/media/pdf/50016527.pdf.

Scheer, A.-W.: ARIS - House of Business Engineering 1996
Scheer, A.-W.: ARIS - House of Business Engineering, in: IDS Prof. Scheer GmbH (Ed.): Scheer magazin spezial, Oktober 1996.

Scheer, ARIS – Business Process Modeling 1998
Scheer, A.-W.: ARIS – Business Process Modeling, 2. ed., Berlin et al. 1998.

Scheer, CIM 1994
Scheer, A.-W.: CIM (Computer Integrated Manufacturing) – Towards the Factory of the Future, 3. ed., Berlin et al. 1994.

Scheer, Data Warehouse 1996
Scheer, A.-W.: Data Warehouse und Data Mining: Konzepte der Entscheidungsunterstützung, in: IM Information Management 11 (1996) 1, pp. 74-75.

Scheer, Efficient Information Management 1991
Scheer, A.-W.: Principles of Efficient Information Management, 2. ed., Berlin et al. 1991.

Scheer, Business Process Engineering 1994
Scheer, A.-W.: Business Process Engineering – Reference Models for Industrial Enterprises, 2. ed., Berlin et al. 1994.

Scheer, Workflow-Systeme 1997
Scheer, A.-W.: Workflow-Systeme: Jetzt auch im Büro, in: Harvard Business Manager 19(1997)1, pp. 115-122.

Scheer/Bold/Hagemeyer/Kraemer, Informationssysteme im Wandel 1997
Scheer, A.-W., Bold, M., Hagemeyer, J., Kraemer, W.: Organisationsstrukturen und Informationssysteme im Wandel - Konsequenzen für die Informationsmodellierung, in: Organisationsstrukturen und Informationssysteme auf dem Prüfstand, 18. Saarbrücker Arbeitstagung, Heidelberg 1997, pp. 3-32.

Schmidt, Informationsmanagement 1996
Schmidt, G.: Informationsmanagement - Modelle, Methoden, Techniken, Berlin et al. 1996.

Schröder, Business Engineer 1997
Schröder, G.: Configure to Order using the R/3 Business Engineer, Vortrag Sapphire, Amsterdam, Juni 1997,
URL: http://www.sap.com/events/amsterdam/sessions/index1.htm.

Schwabe/Krcmar, CSCW-Werkzeuge 1997
Schwabe, G., Krcmar, H.: CSCW-Werkzeuge, in: Wirtschaftsinformatik 38(1997)2, pp. 209-224.

Sebestyén, Management-„Geheimnis" Kaizen 1994
Sebestyén, O.G.: Management-„Geheimnis" Kaizen, Der japanische Weg zur Innovation, Wien 1994.

Seghezzi/Berger, Qualitätsstrategien 1993
Seghezzi, H.D., Berger, R.: Qualitätsstrategien. Anforderungen an das Management der Zukunft, München et al. 1993.

Seghezzi/Dahlem, Schritt für Schritt zu TQM 1997
Seghezzi, H.D., Dahlem, S.: Schritt für Schritt zu TQM. In: QZ 42(1997)5, pp. 553-558.

Shlaer/Mellor, Object Oriented Systems Analysis 1988
Shlaer, S., Mellor, S.J.: Object Oriented Systems Analysis, Englewood Cliffs 1988.

Siemens Nixdorf, ComUnity 1997
Siemens Nixdorf Informationssysteme AG (Ed.): ComUnity Visual Framework-Overview and Architecture 1997,
URL: http://comunity.sni.de/public/cvf/crf_en/info.htm.

Sinz, Modellierung betrieblicher Informationssysteme 1996
Sinz, E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends, in: Heilmann, H., Heinrich, L.J., Roithmayr, F. (Ed.): Information Engineering, München et al. 1996, pp. 123-143.

Sommerville, Software Engineering 1987
Sommerville, I.: Software Engineering, Bonn et al. 1987.

Sol, Information Systems Design Methodologies 1983

Sol, H.G.: A Feature Analysis of Information Systems Design Methodologies: Methodological Considerations, in: Olle, T.W., Sol, H.G., Tully, C.J. (Ed.): Information Systems Design Methodologies: A Feature Analysis, Proceedings of the IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies, Amsterdam et al. 1983, pp. 1-8.

Sowa/Zachman, Framework for Information Systems Architecture 1992

Sowa, F., Zachman, J.A.: Extending and Formalizing the Framework for Information Systems Architecture, in: IBM System Journal, 31 (1992), pp. 590-619.

Spek/Hoog, Knowledge Management 1994

Spek, R.v.d.; de Hoog, R.: Towards a methodology for knowledge management, Technical Note Knowledge Management Network, 23. November 1994, URL: http://ceres.cibit.nl/web/kmn/pospapers.nsf.

Strunz, Informations- und Kommunikationssysteme 1990

Strunz, H.: Informations- und Kommunikationssysteme, Handbuch Wirtschaftsinformatik, Strunz, H., Kurbel, K. (Ed.), Wiesbaden 1990.

Stülpnagel, Repositories 1991

Stülpnagel, A. v.: Repositories - Konzepte, Architekturen, Standards, in: HMD 161(1991)28, pp. 10-25.

Szidzek, Datenmodellierung - Vorgehensmodell 1993

Szidzek, A.: Datenmodellierung - Vorgehensmodell zur Konstruktion und Einführung einer unternehmensweiten konzeptionellen Datenstruktur, in: Thome, R. (Ed.): Forschung und Praxis der Wirtschaftsinformatik, Band 2, Würzburg 1993.

Thome/Hufgard, Continuous System Engineering 1996

Thome, R., Hufgard, A.: Continuous System Engineering, Entdeckung der Standardsoftware als Organisator, Würzburg 1996.

UML Notation Guide 1997

Rational Software et al.: UML Notation Guide, Version 1.1, 01.09.1997, URL: http://www.rational.com/uml/html/notation.

UML Semantics 1997

Rational Software et al.: UML Semantics, Version 1.1, 01.09.1997, URL: http://www.rational.com/uml/html/semantics.

UML Summary 1997

Rational Software et al.: UML Summary, Version 1.1, 01.09.1997, URL: http://www.rational.com/uml/html/summary.

Vernadat, Enterprise Modeling and Integration 1996

Vernadat, F.B.: Enterprise Modeling and Integration: Principles and Applications, London 1996.

Vos, Groupware 1997
Vos, P.: Groupware läßt die Arbeit fließen, in: online 9/97, pp. 40-44.

Wagner, Standardisierungsbestrebungen für Business Objects 1997
Wagner, D.: Standardisierungsbestrebungen für Business Objects, Seminararbeit an der Universität des Saarlandes, Lehrstuhl für Betriebswirtschaftslehre, insb. Wirtschaftsinformatik, Saarbrücken 1997.

Winter/Maag, AD/Cycle 1990
Winter, F., Maag, D.: AD/Cycle - Verstärkung für SAA?, in: IM Information Management, 5 (1990) 2, pp. 32-39.

Williams, T.J. et al.: Purdue Enterprise Reference Architecture 1991
Williams, T.J. et al.: The Purdue Enterprise Reference Architecture, Report Number 154, Purdue Laboratory for Applied Industrial Control, Purdue University, Indiana 1991.

Wirfs-Brock/Wilkerson/Wiener, Objektorientiertes Software-Design 1993
Wirfs-Brock, R., Wilkerson, B., Wiener, L.: Objektorientiertes Software-Design, München 1993.

Wollnick, Referenzmodell des Informationsmanagements 1988
Wollnick, M.: Ein Referenzmodell des Informationsmanagements, in: IM Information Management, 3 (1988) 3, pp. 34-43.

Zachman, Framework for Information Systems Architecture 1987
Zachman, J.A.: A Framework for Information Systems Architecture, in: IBM Systems Journal, 26 (1987) 3, pp. 276-292.

Zell, Führungsinformationssysteme 1997
Zell, M.: Informationstechnische Gestaltung von Führungsinformationssystemen, in: Controlling 9 (1997) 4, pp. 290-301.

Zencke, BAPI 1997
Zencke, P: „Unsere Kunden nutzen BAPIs sehr kreativ für ihre Internet-Anwendungen", Computerzeitung, 44 (1997) 10, p. 14.

Zimmermann, Produktionsfaktor Information 1972
Zimmermann, D.: Produktionsfaktor Information, Neuwied-Berlin 1972.

Zucker/Schmitz, Knowledge Flow Management 1994
Zucker, B.; Schmitz, C.: Knowledge Flow Management: Wissen nutzen statt verspielen, in: Gablers Magazin 8 (1994) 11-12, pp. 62-65.

Index

D

E

F-G

H-J

K-L

M

O

P

Q-R

S

T-V

W-Z